

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of :
Yukie GOTOH et al. :
Serial No. NEW : **Attn: APPLICATION BRANCH**
Filed January 23, 2004 : Attorney Docket No. 2004-0089A

COMMON KEY EXCHANGING METHOD
AND COMMUNICATION DEVICE

CLAIM OF PRIORITY UNDER 35 USC 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

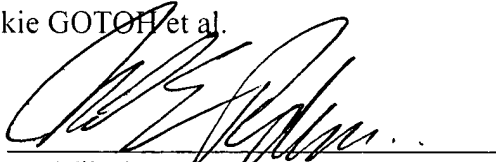
Applicants in the above-entitled application hereby claim the date of priority under the International Convention of Japanese Patent Application No. 2003-015866, filed January 24, 2003, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Yukie GOTOH et al.

By



Nils E. Pedersen

Registration No. 33,145

Attorney for Applicants

NEP/krq
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
January 23, 2004

THE COMMISSIONER IS AUTHORIZED
TO CHARGE ANY DEFICIENCY IN THE
FEES FOR THIS PAPER TO DEPOSIT
ACCOUNT NO. 23-0975

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2003年 1月24日

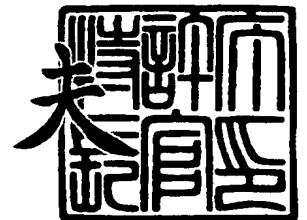
出願番号
Application Number: 特願2003-015866
[ST. 10/C]: [JP 2003-015866]

出願人
Applicant(s): 松下電器産業株式会社

2003年11月28日

特許庁長官
Commissioner,
Japan Patent Office

今井 康夫



出証番号 出証特2003-3098664



【書類名】 特許願

【整理番号】 2032740136

【提出日】 平成15年 1月24日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 9/00

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 五島 雪絵

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 横田 博史

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 玉井 昌朗

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 辻 敦宏

【発明者】

 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

 【氏名】 高垣 景一

【特許出願人】

 【識別番号】 000005821

 【氏名又は名称】 松下電器産業株式会社

**【代理人】****【識別番号】** 100097445**【弁理士】****【氏名又は名称】** 岩橋 文雄**【選任した代理人】****【識別番号】** 100103355**【弁理士】****【氏名又は名称】** 坂口 智康**【選任した代理人】****【識別番号】** 100109667**【弁理士】****【氏名又は名称】** 内藤 浩樹**【手数料の表示】****【予納台帳番号】** 011305**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【包括委任状番号】** 9809938

【書類名】 明細書

【発明の名称】 共有鍵交換方法および通信機器

【特許請求の範囲】

【請求項 1】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 1 の通信機器は、自分が生成した第 1 の秘密情報を基に第 1 の公開値を計算し、前記第 2 の通信機器に前記第 1 の公開値を送信し、

前記第 2 の通信機器は、第 2 の公開値を送信するまでの時間が遅くなることを応答予告として、前記第 1 の公開値の受信前または受信直後に、前記第 1 の通信機器に予め通知しておき、その後、自分が生成した第 2 の秘密情報を基に前記第 2 の公開値を計算し、前記第 1 の通信機器に前記第 2 の公開値を送信し、

前記第 1 の通信機器は、前記第 2 の通信機器から前記応答予告を受け取るようにしたことを特徴とする共有鍵交換方法。

【請求項 2】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 1 の通信機器は、自分が生成した第 1 の秘密情報を基に第 1 の公開値を計算し、前記第 2 の通信機器に前記第 1 の公開値を送信し、

前記第 2 の通信機器は、第 2 の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第 1 の公開値の受信前または受信直後に、前記第 1 の通信機器に予め通知しておき、その後、自分が生成した第 2 の秘密情報を基に前記第 2 の公開値を計算し、前記第 1 の通信機器に前記第 2 の公開値を送信し

前記第 1 の通信機器は、前記第 2 の通信機器から前記応答予告を受け取ると、前記第 2 の通信機器から今後送られてくるはずの前記第 2 の公開値を、少なくとも前記応答予告に基づく時間以上待機することを特徴とする共有鍵交換方法。

【請求項 3】 前記第 1 の通信機器は、共有鍵交換の手順に先だって、前記第 2 の通信機器に前記応答予告を問い合わせるメッセージを発行し、前記第 2 の通信機器は前記メッセージに対して、第 2 の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第 1 の通信機器に送信することを特徴とする、請求項 1 または請求項 2 に記載の共有鍵交換方法。

【請求項 4】 前記第 2 の通信機器は、第 2 の公開値を算出するための処理時間を事前に実測しておき、前記第 1 の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻を算出することを特徴とする、請求項 2 または請求項 3 に記載の共有鍵交換方法。

【請求項 5】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 1 の通信機器は、自分が生成した第 1 の秘密情報を基に第 1 の公開値を計算し、前記第 2 の通信機器に前記第 1 の公開値を送信し、

前記第 1 の通信機器は、タイムアウト時間より小さい再送間隔時間毎に、第 2 の通信機器に対して状況確認を行ない、

前記第 2 の通信機器は、後で送信する旨の通知を、前記第 1 の通信機器に行ない、

前記第 1 の通信機器は、後で送信する旨の通知を受信した場合、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始めるようにしたことを特徴とする共有鍵交換方法。

【請求項 6】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 2 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を前記第 1 の通信機器に通知し、

前記第 1 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間と、通知された前記第 2 の通信機器が公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定し、

前記第 1 の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【請求項 7】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 2 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、

前記第 2 の通信機器は、前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、

前記第 2 の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、前記第 1 の通信機器に、共有鍵交換処理の開始を要求することを特徴とする共有鍵交換方法。

【請求項 8】 前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために自身に必要な時間の 2 倍以上前であることを特徴とする請求項 7 に記載の共有鍵交換方法。

【請求項 9】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 1 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

前記第 2 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

前記第 1 の通信機器と第 2 の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処理の開始を通知し、

前記第 1 の通信機器と第 2 の通信機器は、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【請求項 10】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 1 の通信機器は、自分が生成した第 1 の秘密情報を基に第 1 の公開値を計算し、前記第 2 の通信機器に前記第 1 の公開値を送信し、

前記第 2 の通信機器は、前記第 1 の通信機器から送られた第 1 の公開値が、既に存在する鍵を更新するための要求か否かを判定し、

鍵を更新するための要求と判定した場合には、自分が生成した第 2 の秘密情報を基に前記第 2 の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第 1 の通信機器に前記第 2 の公開値

を送信することを特徴とする共有鍵交換方法。

【請求項 11】 前記第 2 の通信機器は、第 2 の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出して、前記第 1 の通信機器に通知しておく、

前記第 1 の通信機器は、前記第 2 の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出し、前記タイミングの時点までに共有鍵交換を開始することを特徴とする、請求項 10 に記載の共有鍵交換方法。

【請求項 12】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 2 の通信機器は、次回の共有鍵交換処理が、既に存在する鍵を更新するための処理か否かを判定し、

鍵を更新するための処理と判定した場合には、自身が第 2 の秘密情報を基に前記第 2 の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第 1 の通信機器に前記第 2 の公開値を送信することを特徴とする共有鍵交換方法。

【請求項 13】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 1、または、第 2 の通信機器は、共有鍵の演算を終了後、演算終了の通知を、相手の通信機器に行ない、相手の通信機器は、演算終了の通知を受けるまで、タイムアウト処理を行なわないようにすることを特徴とする共有鍵交換方法

。

【請求項 14】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、

前記 2 つの通信機器は、第 1 の通信機器と第 2 の通信機器から成り、

前記第 1 の通信機器および前記第 2 の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、

前記第 2 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

前記第 1 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする共有鍵交換方法。

【請求項 15】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第 1 の秘密情報を基に第 1 の公開値を計算し、第 2 の通信機器に第 1 の公開値を送信し、

第 2 の通信機器が行なう、第 2 の公開値を計算し送信するまでの時間が遅くなることを応答予告とする通知を、受信することを特徴とする第 1 の通信機器。

【請求項 16】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第 1 の秘密情報を基に第 1 の公開値を計算し、第 2 の通信機器に第 1 の公開値を送信し、

第 2 の通信機器が、第 2 の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、第 1 の公開値の受信前または受信直後に、予め行なう通知

を受信しておき、その後、第2の通信機器自身が生成した第2の秘密情報を基に計算した第2の公開値を受信し

第2の通信機器から前記応答予告を受け取ると、第2の通信機器から今後送られてくるはずの第2の公開値を、少なくとも前記応答予告に基づく時間以上、受信できる状態で待機することを特徴とする第1の通信機器。

【請求項17】 前記第1の通信機器は、共有鍵交換の手順に先だって、前記第2の通信機器に前記応答予告時間を問い合わせるメッセージを発行し、前記第2の通信機器が応答メッセージとして送信する第2の公開値を送信するまでの遅延時間または送信予定時刻を、前記応答予告として受信することを特徴とする、請求項15または請求項16に記載の第1の通信機器。

【請求項18】 前記第2の通信機器に、第2の公開値を算出するための処理時間を事前に実測させておき、前記第1の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻を算出させることを特徴とする、請求項16または請求項17に記載の第1または第2の通信機器。

【請求項19】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第1の秘密情報を基に第1の公開値を計算し、第2の通信機器に第1の公開値を送信し、

タイムアウト時間より小さい再送間隔時間毎に、第2の通信機器に対して状況確認を行ない、

第2の通信機器が送信する、後で送信する旨の通知を、受信し、

後で送信する旨の通知を受信した場合、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始めるようにしたことを特徴とする第1の通信機器。

【請求項20】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信

した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第2の通信機器が通知する、公開値の生成と共有鍵の生成のために自身に必要な時間を受信し、

公開値の生成と共有鍵の生成のために自身に必要な時間と、通知された前記第2の通信機器が公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定し、

前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始することを特徴とする第1の通信機器。

【請求項21】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第2の通信機器が、公開値の生成と共有鍵の生成のために第2の通信機器自身に必要な時間を算出し、前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、前記共有鍵交換処理を開始すべき時刻までに、前記第2の通信機器から、共有鍵交換処理の開始を要求された場合、共有鍵交換処理を開始することを特徴とする第1の通信機器。

【請求項22】 前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために第2の通信機器自身に必要な時間の2倍以上前であることを特徴とする請求項21に記載の第1の通信機器。

【請求項23】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第1の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

第2の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算出し、

第1の通信機器と第2の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処理の開始を通知し、共有鍵交換処理の開始を通知された場合、共有鍵交換処理を開始することを特徴とする第1の通信機器。

【請求項24】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

自身が生成した第1の秘密情報を基に第1の公開値を計算し、前記第2の通信機器に前記第1の公開値を送信し、

第2の通信機器において、第2の通信機器が受信した第1の公開値が、すでに存在する鍵を更新するための要求か否かを判定し、鍵を更新するための要求と判定された場合には、第2の通信機器自身が生成した第2の秘密情報を基に前記第2の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行することにより第2の通信機器において生成された、前記第2の公開値を受信することを特徴とする第1の通信機器。

【請求項25】 前記第2の通信機器において、第2の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出したものを受信し、

前記第2の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出し、前記タイミングの時点までに共有鍵交換を開始することを特徴とする、請求項24に記載の第1の通信機器。

【請求項26】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第2の通信機器において、次の共有鍵交換処理が、既に存在する鍵を更新す

るための処理か否かを判定し、鍵を更新するための処理と判定した場合には、第 2 の通信機器自身が第 2 の秘密情報を基に第 2 の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、第 2 の通信機器において生成された前記第 2 の公開値を受信することを特徴とする第 1 の通信機器。

【請求項 2 7】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

相手から共有鍵の演算終了の通知を受けるまで、タイムアウト処理を行わないようにすることを特徴とする第 1 の通信機器。

【請求項 2 8】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第 2 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

第 1 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする第 1 の通信機器。

【請求項 2 9】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第 2 の公開値を送信するまでの時間が遅くなることを応答予告として第 1 の通信機器に対して通知することを特徴とする第 2 の通信機器。

【請求項 3 0】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信

した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第1の通信機器において生成した第1の公開値を受信し、

第2の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第1の公開値の受信前または受信直後に、前記第1の通信機器に予め通知しておき、その後、自身が生成した第2の秘密情報を基に前記第2の公開値を計算し、第1の通信機器に前記第2の公開値を送信し、

第1の通信機器に、今後送る予定である前記第2の公開値の送信を、少なくとも前記応答予告に基づく時間以上待たせることを特徴とする第2の通信機器。

【請求項31】 前記第1の通信機器が、共有鍵交換の手順に先だって、前記第2の通信機器に前記応答予告を問い合わせるメッセージに対する応答メッセージとして第2公開値を送信するまでの遅延時間または送信予定時刻を前記応答予告として前記第1の通信機器に送信することを特徴とする、請求項29または請求項30に記載の第2の通信機器。

【請求項32】 前記第2の通信機器は、第2の公開値を算出するための処理時間を事前に実測しておき、前記第1の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻時間を算出することを特徴とする、請求項30または請求項31に記載の第2の通信機器。

【請求項33】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第1の通信機器が生成した第1の公開値を受信し、

第1の通信機器が、タイムアウト時間より小さい再送間隔時間毎に、行なう状況確認を受信し、状況確認に対して、後で送信する旨の通知を、第1の通信機器に行ない、

第1の通信機器に、前記後で送信する旨の通知により、タイムアウト時間をリセットさせ、新たにタイムアウト時間の計測を始めさせるようにしたことを特徴

とする第 2 の通信機器。

【請求項 3 4】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

公開値の生成と共有鍵の生成のために自身に必要な時間を前記第 1 の通信機器に通知し、

第 1 の通信機器に、公開値の生成と共有鍵の生成のために第 1 の通信機器自身に必要な時間と、通知した前記公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定させ、

前記第 1 の通信機器に、前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始させることを特徴とする第 2 の通信機器。

【請求項 3 5】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、

前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、

前記共有鍵交換処理を開始すべき時刻までに、第 1 の通信機器に、共有鍵交換処理の開始を要求することを特徴とする第 2 の通信機器。

【請求項 3 6】 前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために自身に必要な時間の 2 倍以上前であることを特徴とする請求項 3 5 に記載の第 2 の通信機器。

【請求項 3 7】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第 1 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し

第 2 の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、

第 1 の通信機器と第 2 の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処理の開始を通知し、

通知された場合、共有鍵交換処理を開始することを特徴とする第 2 の通信機器。

【請求項 3 8】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第 1 の通信機器により生成された第 1 の公開値を受信し

第 1 の通信機器から送られた第 1 の公開値が、既に存在する鍵を更新するための要求か否かを判定し、

鍵を更新するための要求と判定した場合には、自分が生成した第 2 の秘密情報を基に前記第 2 の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第 1 の通信機器に前記第 2 の公開値を送信することを特徴とする第 2 の通信機器。

【請求項 3 9】 前記第 2 の通信機器は、第 2 の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出して、前記第 1 の通信機器に通知しておき、

前記第 1 の通信機器に、前記第 2 の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出させ、前記タイミングの時点までに共有鍵交換を開始させることを特徴とする、請求項 3 8 に記載の第 2 の通信機器。

【請求項 4 0】 暗号・認証処理を施したデータを送受信する 2 つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法

に使用する通信機器であって、

次回の共有鍵交換処理が、既に存在する鍵を更新するための処理か否かを判定し、

鍵を更新するための処理と判定した場合には、自身が第2の秘密情報を基に前記第2の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、第1の通信機器に前記第2の公開値を送信することを特徴とする第2の通信機器。

【請求項41】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

相手から共有鍵の演算終了の通知を受けるまで、タイムアウト処理を行わないようにすることを特徴とする第2の通信機器。

【請求項42】 暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法に使用し、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算する共有鍵交換方法に使用する通信機器であって、

第2の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

第1の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、

上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする第2の通信機器。

【請求項43】 請求項1～12に記載の処理を、コンピュータに機能させるためのプログラムとして記録した共有鍵交換方法プログラム記録媒体。

【請求項44】 請求項1～12に記載の処理を、コンピュータに機能させるためのプログラムとした共有鍵交換方法のプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ネットワークを介して秘匿性の高いデータを送受信する機器間で、暗号および認証のための秘密鍵を交換・共有するための共有鍵交換方法の技術分野に関する。

【0002】

【従来の技術】

近年のインターネットの普及により、電子メール、電子商取引など、ネットワークを介したサービスも増大し、家庭内にもネットワークに接続できる機器が増えつつある。このネットワークに接続できる家電、いわゆる「ネット家電」としては、エアコンや電子レンジなどの白物家電もインターネットに接続し、宅外からエアコンを制御したり、センター側からプログラムのバージョンアップを行うようなサービスも提案されている。

【0003】

しかしながら、このようなネットワークの拡大に伴い、ネットワーク上の電子化されたデータを改ざんしたり、盗聴／盗み見したり、他人になりすましてサービスを受けたりする、といった問題が浮上しており、ネットワーク上のセキュリティ対策が重要となってきている。

【0004】

ネットワーク上のセキュリティ機能を実現する取り組みとしては、暗号化技術や認証技術を使ったプロトコルが、複数、規格化・実用化されている。例えば、IPパケットのレベルでのセキュリティ機能を規定したプロトコルとしては、IPsec (IP Security protocol) がある。

【0005】

IPsecとは、IETFによって標準化されたプロトコルであり、内容は、非特許文献1 (RFC 2401) などによって規定されている。IPsecでは、IPパケットそのものを暗号化することで盗聴者に対して通信内容を保護し、秘密の漏洩を防ぐことができる。また、IPパケットに認証（完全性チェック）用の値を追加することで、通信経路でデータが改ざんされなかったことを保証する。

【0006】

一方、IPsecなどのプロトコルを使って、機器間で暗号・認証処理されたデータを送受信する場合、データ送受信に先だって、暗号・認証処理に用いる鍵（以下、セッション共有鍵と呼ぶ）を両方の機器で共有する必要がある。ただし、ユーザが手動でセッション共有鍵を設定するのは煩雑であり、多くの機器間で通信するシステムには向かない。さらにセッション共有鍵を長時間、大量のデータに対して使いつづけると、第三者にセッション共有鍵を解読されてしまう可能性が高まるため、必要に応じて鍵を更新する必要がある。そのため、鍵を自動的に生成、配布、交換、更新するためのさまざまな方法が提案されている。

【0007】

例えば、2つの機器間で1つの秘密の対称型鍵を交換させる方法として、広く使われている基本技術に、Diffie-Hellman（ディフィーヘルマン）の方法がある。この方法は、前述のIPsecの鍵交換・管理プロトコルとして、標準使用が規定されているIKE（Internet Key Exchange）にも使われている。非特許文献2（RFC2407～2409）に開示されているIKEの概要は、後述する。これらの鍵交換方法は、いかに安全に（第三者に鍵を推察されずに）、かつ、通信をしたい機器同士がいかに簡単に鍵を交換／共有するか、が課題である。

【0008】

以下、従来の共有鍵交換（共有、配送）方法の具体例として、Diffie-Hellmanの鍵交換方法を説明する。

【0009】

図11は、非特許文献3（RFC2631）で規定されているDiffie-Hellmanの鍵交換方法を説明するための図である。なお、図中で、S付きの数字は、処理のステップを表している。図中で機器Aと機器Bは、暗号通信と鍵交換を行う機器の組とする。また、機器Aと機器Bは鍵交換を行う前に、 g と n という変数の値を知っているものとする（ステップ1101、ステップ1102）。ただし、 g と n の値は、第三者（盗聴者）が知っていても良い。

【0010】

まず、機器Aは秘密の値 a を生成し（ステップ1103）、秘密値 a を元に公開値 X を算出する（ステップ1104）。公開値 X は以下の計算式で求める。

【0011】

【数1】

$$X = g^a \bmod n$$

【0012】

ここで、上式は、 g の a 乗を n で割った余りを公開値 X とすることを意味する。算出した公開値 X は、機器Bに送る（ステップ1105）。

【0013】

一方、機器Bは機器Aから公開値 X を受け取ると、（ステップ1103）、（ステップ1104）と同様に、機器B自身の秘密値 b の生成（ステップ1106）、公開値 Y の算出（ステップ1107）を行う。

【0014】

【数2】

$$Y = g^b \bmod n$$

【0015】

さらに機器Bでは機器Aからの公開値 X と機器B自身の秘密値 b を元に、以下の式で共有鍵 K を算出する（ステップ1108）。

【0016】

【数3】

$$K = X^b \bmod n$$

【0017】

最後に機器Bは、自分の公開値 Y を機器Aに送る（ステップ1109）。機器Aでは、機器Bから公開値 Y を受け取ると、機器Bからの公開値 Y と機器A自身の秘密値 a を元に、以下の式で共有鍵 K を算出する（ステップ1110）。

【0018】

【数4】

$$K = Y^a \bmod n$$

【0019】

(数3)、(数4)では、どちらの式も(数5)の関係が成立する。

【0020】

【数5】

$$K = X^b \bmod n = g^{(a \times b)} \bmod n$$

$$K = Y^a \bmod n = g^{(a \times b)} \bmod n$$

【0021】

そのため、機器A、機器Bは、共有鍵Kとして同じ値を算出することができる。両方の機器で共有鍵Kを算出する、すなわち共有鍵Kを共有した後は、共有鍵Kを用いて暗号／認証処理したデータを、機器間で送受信することができる（ステップ1111）。

【0022】

一方、盗聴者が既知の g 、 n 、および公開値 X 、 Y を基に共有鍵 K を見つけ出すことは、いわゆる離散対数問題といわれ、大変難しい問題であると知られている。特に n を極めて大きく（数百ビットから数千ビット）に設定した場合には、離散対数を解くことは事実上不可能である。

【0023】

以上のように、Diffie-Hellmanの鍵交換の手順を使うことにより、機器A－機器B間で安全に共有鍵 K を共有させることができる。

【0024】

なお、前述したIKEの場合には、Diffie-Hellmanの鍵交換の手順だけでなく、その前後に機器間でやりとりする手順も規定されている。以下、図12を使ってIKEでの鍵交換方法を簡単に説明する。なおIKEにおいては、最初に鍵交換の要求を相手に送る側の機器をイニシエータと呼び、その要求を受け付ける側の機器をレスポндаと呼ぶ。図中の左側の処理フローはイニシエータ側の機器に対応し、右側の処理フローはレスポнда側の機器に対応する。また、IKEではPhase 1（ステップ1201）とPhase 2（ステップ1205）という2段階の鍵交換を規定している。

【0025】

まずPhase 1の段階（ステップ1201）において、イニシエータとレス

ポンドは、はじめに鍵交換と暗号／認証処理に用いる各種パラメータを折衝する（ステップ1202）。例えば、Diffie-Hellmanで用いる g と n の値や、後の（ステップ1204）で用いる暗号化および認証のアルゴリズムもここで決められる。 g と n の値については、値そのもの、あるいは、 g と n の値を特定するグループ番号の形で取り決める。次に、前述したDiffie-Hellmanの鍵交換を行い、共有鍵 K_Phase1 を共有する（ステップ1203）。 $Phase1$ の最後には、イニシエータとレスポンドが相互に本当に正しい相手かどうかを認証するための手順を行う（ステップ1204）。このステップ1204の本人性認証で使うメッセージは、共有鍵 K_Phase1 に基づく鍵によって暗号／認証処理した後、送受信する。お互いの機器を認証するための情報としては、IDやハッシュ値などを使うことができる。たとえば、共有鍵 K_Phase1 とステップ1203で交換したデータなどを連結し、それを鍵つきハッシュ関数の入力値として計算したハッシュ値を、ステップ1204のメッセージの暗号／認証に使う鍵として使用できる。

【0026】

次の $Phase2$ の段階（ステップ1205）は、イニシエータ・レスポンド間で実際に送受信したいデータに対する暗号化／認証処理について決定する場合である。 $Phase2$ のメッセージでは、ステップ1204と同様、共有鍵 K_Phase1 に基づく鍵で暗号化および認証処理が施される。 $Phase2$ の最初の2メッセージでは、 g 、 n の値の提案又は通知、暗号化および認証のアルゴリズムなどの折衝と、Diffie-Hellmanの公開値の交換を行う（ステップ1206）。 $Phase2$ の（ g 、 n 、 a 、 b ）には、 $Phase1$ のパラメータと異なる値を使うことができる。イニシエータとレスポンドは、ステップ1206のDiffie-Hellman交換で計算される共有鍵 K_Phase2 を基に、実際のデータに適用する暗号・認証処理の鍵を生成し、暗号化・認証処理済みのデータを送受信できるようになる（ステップ1207）。

【0027】

IKEでは、上記手順以外にも、一方のメッセージが他方の機器に届かなかった場合の対策であるメッセージの再送や、鍵の更新手順なども規定されている。

【0028】

【非特許文献1】

「RFC2401」、IETF (Internet Engineering Task Force) 発行

【非特許文献2】

「RFC2407」～「RFC2409」、IETF (Internet Engineering Task Force) 発行

【非特許文献3】

「RFC2631」、IETF (Internet Engineering Task Force) 発行

【0029】

【発明が解決しようとする課題】

上記のように、従来提案されてきたDiffie-Hellmanなどの共有鍵交換法を使うことで、通信を行う機器間で鍵を安全に共有させることができる。

【0030】

しかしながら、Diffie-Hellmanの共有鍵交換法はべき乗演算とモジュロ演算を含んでおり、桁数の大きい数値を使う場合の処理負荷は非常に重いことが知られている。例えば、IKEで規定されているDHグループ2の場合には、 $g=2$ 、 $n:1024$ ビットの定数を使って公開値X、Yおよびセッション共有鍵Kを計算する。特にネット家電のように低コストのCPUを搭載する機器では、高い計算力が望めず、以下のような3つの課題が発生する。

【0031】

- ・第1の課題：対向の機器でのタイムアウト検出により、鍵交換が失敗する。

【0032】

- ・第2の課題：鍵の寿命が設定されている場合、寿命満了までに、鍵の更新が完了しない。

【0033】

- ・第3の課題：機器上の他のアプリケーションの実行を妨害する。

【0034】

(第1の課題) 対向の機器でのタイムアウト検出による鍵交換の失敗

以下、図13を使って第1の課題が発生する具体例を説明する。図13は、機器Aと機器Bの間でDiffie-Hellmanの鍵交換を行うシーケンスを示したものであり、機器Bは低い性能の(計算力の低い)CPUを搭載する機器とする。

【0035】

まず、図13のステップ1301~1306は、図11の1101~1106と同じであり、機器A側からDiffie-Hellmanの公開値Xを送る。次に、機器Bは図11と同様に公開値Yとセッション共有鍵Kの演算を行うが(ステップ1307、1308)、CPUの計算力が低いため、非常に長い処理時間を要する。

【0036】

機器A側では、自分の公開値Xを送った後(ステップ1305)、対向の機器(図13では機器B)からの公開値Yの応答を待つが、通常、通信経路中のパケットロスや対向機器の障害などの事態に備えて、再送やタイムアウトの機構を持たせている。図13の機器Aの場合は、機器Bからの応答が無ければ再送間隔= T [秒] の時間毎に公開値Xのメッセージを再送する(ステップ1310)。さらに再送を3回繰り返し(ステップ1311、1312)、それでも機器Bからの応答が無い場合には、 $T \times 4$ [秒] 後にタイムアウトを検出して、鍵交換が失敗したとみなす(ステップ1313)。

【0037】

その後、聞きBで公開値Yとセッション共有鍵Kの演算が終了し、公開値Yを送った(ステップ1309)としても、そのときには機器Aでは既に機器Bとの鍵交換処理中の状態をリセットしている。そのため、公開値Yを含むメッセージは、無効なメッセージと判断して廃棄されてしまう(ステップ1314)。

【0038】

(第2の課題) 鍵更新の完了前に、古い鍵の寿命が切れる課題

同一の鍵を長時間にわたって使用すると、その間に鍵を解読されることが全く

ないとは言えない。そこで、IKEでは、鍵に寿命を設けておき、適当な時間が経つと、それまでの鍵の使用を終了し、別の新しい鍵を使用することができる。具体的には、鍵交換成功により、鍵が作成されると、その鍵に寿命を設定する。あるいは、鍵交換の手順を開始する際に、その鍵に付与する寿命満了の時刻を機器A、B間で決めておく。その鍵の寿命が満了になる前に、次の新しい鍵を作成し、使用中の鍵の寿命が満了となる前に、この新しい鍵に切り替える。このような鍵の更新処理を、リキー処理と呼ぶ。Diffie-Hellmanに伴う演算に時間がかかると、リキー処理が遅れ、古い鍵の寿命満了までに鍵の更新ができず、古い鍵を使い続けるうちに、その寿命が満了するとパケットの暗号化ができなくなる課題があった。

【0039】

(第3の課題) 他のアプリケーションへの妨害

Diffie-Hellmanに伴う演算を行うと、長時間にわたってネット家電のCPUをこの演算だけで占有してしまい、CPUリソースが枯渇する。そのため、ネット家電上で動作する他のアプリケーションの実行が妨げられ、正常に動作しなくなる場合がある。

【0040】

それゆえ、本発明の第1の目的は、低性能のCPUを使う場合にも、対向の機器でタイムアウトを検出させずに鍵交換を成功させるような共有鍵交換方法を提供することである。本発明の第2の目的は、低性能のCPUを使う場合にも、リキーが正常に行なわれるような共有鍵交換方法を提供することである。さらに本発明の第3の目的は、低性能のCPUを使う場合にも、同じCPU上の他のアプリケーションの実行を妨げることなく、同時に鍵交換も成功させるような共有鍵交換方法を提供することである。

【0041】

【課題を解決するための手段】

本発明では、上記課題を解決するために、以下のような、方法、および、手段を適用する。

【0042】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第1の通信機器は、自分が生成した第1の秘密情報を基に第1の公開値を計算し、前記第2の通信機器に前記第1の公開値を送信し、前記第2の通信機器は、第2の公開値を送信するまでの時間が遅くなることを応答予告として、前記第1の公開値の受信前または受信直後に、前記第1の通信機器に予め通知しておき、その後、自分が生成した第2の秘密情報を基に前記第2の公開値を計算し、前記第1の通信機器に前記第2の公開値を送信し、前記第1の通信機器は、前記第2の通信機器から前記応答予告を受け取るようにしたことを特徴とする共有鍵交換方法。

【0043】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第1の通信機器は、自分が生成した第1の秘密情報を基に第1の公開値を計算し、前記第2の通信機器に前記第1の公開値を送信し、前記第2の通信機器は、第2の公開値を送信するまでの遅延時間又は送信予定時刻を応答予告時間として、前記第1の公開値の受信前または受信直後に、前記第1の通信機器に予め通知しておき、その後、自分が生成した第2の秘密情報を基に前記第2の公開値を計算し、前記第1の通信機器に前記第2の公開値を送信し、前記第1の通信機器は、前記第2の通信機器から前記応答予告時間を受け取ると、前記第2の通信機器から今後送られてくるはずの前記第2の公開値を、少なくとも前記応答予告に基づく時間以上待機することを特徴とする共有鍵交換方法。

【0044】

前記第1の通信機器は、共有鍵交換の手順に先だって、前記第2の通信機器に

前記応答予告時間を問い合わせるメッセージを発行し、前記第2の通信機器は前記メッセージに対して、第2の公開値を送信するまでの遅延時間または送信予定時刻を応答予告として、前記第1の通信機器に送信することを特徴とする、請求項1または2に記載の共有鍵交換方法。

【0045】

前記第2の通信機器は、第2の公開値を算出するための処理時間を事前に実測しておき、前記第1の通信機器に前記応答予告を通知する際には、記録しておいた実測の処理時間を基に前記応答予告である遅延時間または送信予定時刻を算出することを特徴とする、請求項2または3に記載の共有鍵交換方法。

【0046】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第1の通信機器は、自分が生成した第1の秘密情報を基に第1の公開値を計算し、前記第2の通信機器に前記第1の公開値を送信し、前記第1の通信機器は、タイムアウト時間より小さい再送間隔時間毎に、第2の通信機器に対して状況確認を行ない、前記第2の通信機器は、後で送信する旨の通知を、前記第1の通信機器に行ない、前記第1の通信機器は、後で送信する旨の通知を受信した場合、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始めるようにしたことを特徴とする共有鍵交換方法。

【0047】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第2の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を前記第1の通信機器に通知し、前記第1の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間と、通知された

前記第2の通信機器が公開値の生成と共有鍵の生成に必要な時間とから、共有鍵交換処理を開始すべき時刻を算定し、前記第1の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【0048】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第2の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記第2の通信機器は、前記必要な時間から、共有鍵交換処理を開始すべき時刻を算定し、前記第2の通信機器は、前記共有鍵交換処理を開始すべき時刻までに、前記第1の通信機器に、共有鍵交換処理の開始を要求することを特徴とする共有鍵交換方法。

【0049】

前記共有鍵交換処理を開始すべき時刻は、共有鍵の生成を完了すべき時刻よりも、前記公開値の生成と共有鍵の生成のために自身に必要な時間の2倍以上前であることを特徴とする請求項7に記載の共有鍵交換方法。

【0050】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第1の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、前記第2の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算出し、前記必要な時間から、自身が共有鍵交換処理を開始すべき時刻を算定し、前記第1の通信機器と第2の通信機器のうち、前記共有鍵交換処理を開始すべき時刻が早い方が、遅い方に共有鍵交換処

理を開始し、前記第1の通信機器と第2の通信機器は、共有鍵交換処理を開始することを特徴とする共有鍵交換方法。

【0051】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第1の通信機器は、自分が生成した第1の秘密情報を基に第1の公開値を計算し、前記第2の通信機器に前記第1の公開値を送信し、前記第2の通信機器は、前記第1の通信機器から送られた第1の公開値が、既に存在する鍵を更新するための要求か否かを判定し、鍵を更新するための要求と判定した場合には、自分が生成した第2の秘密情報を基に前記第2の公開値を計算する処理手順を小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第1の通信機器に前記第2の公開値を送信することを特徴とする共有鍵交換方法。

【0052】

前記第2の通信機器は、第2の公開値を時間的に負荷分散させて算出する際の処理時間を事前に算出して、前記第1の通信機器に通知しておき、前記第1の通信機器は、前記第2の通信機器から通知された前記処理時間を基に、鍵の更新のために共有鍵交換の処理を開始するタイミングを算出し、前記タイミングの時点までに共有鍵交換を開始することを特徴とする、請求項10に記載の共有鍵交換方法。

【0053】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第2の通信機器は、次の共有鍵交換処理が、既に存在する鍵を更新するための処理か否かを判定し、鍵を更新するための処

理と判定した場合には、自身が第2の秘密情報を基に前記第2の公開値を計算する処理手順と共有鍵を生成する処理手順とを、小単位に分割し、さらに小単位の処理を時間的に負荷分散させて実行し、前記第1の通信機器に前記第2の公開値を送信することを特徴とする共有鍵交換方法。

【0054】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第1、または、第2の通信機器は、共有鍵の演算を終了後、演算終了の通知を、相手の通信機器に行ない、相手の通信機器は、演算終了の通知を受けるまで、タイムアウト処理を行なわないようにすることを特徴とする共有鍵交換方法。

【0055】

暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものとし、前記第2の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、前記第1の通信機器は、公開値の生成と共有鍵の生成のために自身に必要な時間を算定し、上記、必要な時間の少なくとも一方の値を基に、共有鍵の寿命の値を決定することを特徴とする共有鍵交換方法。

【0056】

上記請求項1～14に記載の処理を行なう、上記第1の通信機器、または、第2の通信機器。

【0057】

上記請求項1～14に記載の処理を、コンピュータに機能させるためのプログラムを記録した共有鍵交換方法プログラム記録媒体。

【0058】

上記請求項1～14に記載の処理を、コンピュータに機能させるためのプログラムとした共有鍵交換方法のプログラム。

【0059】

【発明の実施の形態】

図2は、本発明の共有鍵交換方法が適用されるネットワーク構成を示す図である。本発明の共有鍵交換方法が適用される実装形態は、ルータやゲートウェイ（GW）の機器に実装されるケース（図2（a））と、端末に実装されるケース（図2（b））の2種類に分類できる。

【0060】

まず、図2（a）は、GW型機器A201およびGW型機器B202に本発明の共有鍵交換処理が実装されるケースを示す。図2（a）において、GW型機器A201とGW型機器B202は、公衆網207を介して接続されている。また、GW型機器A201は、自身のLAN208内の端末である端末A-1 203、端末A-2 204とも接続されている。同様にGW型機器B202は、自身のLAN209内の端末である端末B-1 205、端末B-2 206とも接続されている。

【0061】

LAN208内の端末とLAN209内の端末の間は、GW型機器A201とGW型機器B202を中継して通信でき、GW型機器A201とGW型機器B202との間のデータは暗号化・認証処理される。すなわち、暗号化・認証処理は2つのGW型機器間で終端しており、2つのGW型機器は、それぞれ自身のLAN端末（LAN208内の機器とLAN209内の機器）からのデータを暗号化・認証処理して対向のGWに送信し、対向のGWからの暗号化・認証処理後のデータを受信すると、復号化・認証処理した後それぞれ自身のLAN端末にフォワードする。また、共有鍵の交換処理もGW型機器A201とGW型機器B202の間で終端する。

【0062】

一方、図2（b）では、ホスト型機器A210およびホスト型機器B211に

本発明の共有鍵交換処理が実装されるケースを示す。図2 (b)において、ホスト型機器B211は、公衆網207を介してGW型機器A201またはホスト型機器A210と接続されている。GW型機器A201と自身のLAN208、LAN内の端末A-1 203、端末A-2 204は、図2 (a)と同じである。

【0063】

ホスト型機器BとLAN208内の端末との間は、GW型機器A201を中継して通信でき、ホスト型機器B211とGW型機器A201と間のデータは暗号化・認証処理される。すなわち、ホスト型機器B211は、GW型機器A201と異なり、自身内部の発着データを自分で暗号化・認証処理する。さらにホスト型機器B211は、同様の機能を持つホスト型機器A210との間で、暗号化・認証処理済みのデータを通信できる。また、共有鍵の交換処理に関しては、ホスト型機器B211-GW型機器A201の間、およびホスト型機器B211-ホスト型機器A210の間で、それぞれ、独立で終端する。

【0064】

次に、本発明の共有鍵交換方法が適用される機器の機能ブロックについて図3を使って説明する。図3は、図2で説明したGW型機器B202の機能ブロックを示す。図3のGW型機器B202において、共有鍵交換部301、データベース管理部302、データベース部303、暗号化／認証処理部304、通信プロトコル処理部305、306、LAN I/F307、WAN I/F308とを備える。

【0065】

GW型機器B202は、WAN I/F308を介して公衆網に接続しており、GW型機器A201と通信できる。一方、LAN209とはLAN I/F307を介して接続しており、LAN内の端末205、206と通信できる。通信プロトコル処理部305、306は、IPレイヤ、TCPレイヤなどの通信プロトコルを処理する部分であり、LAN内の端末と公衆網を介した機器との間の送受信パケットのルーティング処理なども行う。

【0066】

共有鍵交換部 301 は、本発明の共有鍵交換方法に基づく処理を行う処理部である。共有鍵交換部 301 では、対向の機器である GW 型機器 A 201 と鍵交換用の通信を行い、GW 型機器 A 201 と共有鍵を共有する。共有鍵は、データベース管理部 302 を介してデータベース部 303 に記録される。

【0067】

データベース部 303 は、暗号／認証処理に用いる共有鍵を保存する記録部であり、鍵だけでなく、暗号アルゴリズムなどの情報を保存する。データベース部 303 への情報の登録、削除などの処理は、データベース管理部 302 が行う。暗号化／認証処理部 304 は、データベース部 303 の鍵情報等を参照して、パケットの暗号化・復号化／認証（完全性チェック）の処理を行う。具体的には、LAN 209 内の端末 B-1 205、または、端末 B-2 206 から LAN 209 内の端末 A-1 203、または、端末 A-2 204 への送信パケットは、平文（暗号化されていない通常のパケット）で受け取る。LAN I/F 307、通信プロトコル処理部 305 を介して暗号化／認証処理部 304 が上記パケットを受け取ると、データを暗号化、認証処理用のデータの追加、などの処理を行い、通信プロトコル処理部 306、WAN I/F 308 を介して公衆網に送出する。暗号化／認証処理部 304 以降の通信プロトコル処理部 306、WAN I/F 308 を介して送信されるパケットは、暗号化済みとなる。また、公衆網側から到着した暗号化済みのパケットの場合も同様に、暗号化／認証処理部 304 で復号化の処理を行い、LAN 209 内の端末に対して平文のデータを中継する。

【0068】

（実施の形態 1）

以下、本発明の実施の形態 1 に係る共有鍵交換方法について説明する。実施の形態 1 は、従来の共有鍵交換方法における 3 つの課題のうち、第 1 の課題を解決するための方法である。

【0069】

図 1 は、本発明の実施の形態 1 に係る共有鍵交換方法の概要を説明するための図である。図 1 において、機器 A と機器 B は、暗号・認証処理と共有鍵交換を終

端する機器の組とする。具体的には、図2中の機器間で共有鍵交換の処理を終端する3つの組み合わせ、すなわち(1)GW型機器A201とGW型機器B202、(2)GW型機器A201とホスト型機器B211、(3)ホスト型機器A210とホスト型機器B211のいずれかの組み合わせに対応するものとする。また、図1中の機器Bは、機器Aに比べて低い性能のCPUを搭載した機器とする。また、図1の機器Aまたは機器Bに関する処理シーケンスは、図3における共有鍵交換部301で行う処理に対応するものとする。

【0070】

まず、図1のステップ101～105は、図9のステップ901～905と同じであり、機器Aは、算出したDiffie-Hellmanの公開値Xを機器Bに送る。次に、機器Bはこれから自分が計算すべき公開値Yと共有鍵Kの演算所要時間の推定値： T_b を求める(ステップ106)。具体的には、 n と g の取り得る値のさまざまな組み合わせに関して、事前に機器B上で演算を複数回実行しておき、その最大値 $Time_max[g, n]$ を保存しておく。この値に、予め決めておいた固定値： α =機器A/機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値、を足して、以下の式で推定値 T_b を求める。

【0071】

【数6】

$$T_b = Time_max[g, n] + \alpha$$

【0072】

機器Bは、算出した T_b を機器Aに送り、 T_b 時間以内に公開値Yを応答することを予告する(ステップ107)。機器Aは、 T_b を受け取ると、機器Bからの応答待ち期限を T_b 以上に延長する(ステップ111)。すなわち、機器Aでは機器Bからの公開値Yを T_b の時間以上待つように制御することで、機器Bから最大 T_b の時間、応答が無くても、タイムアウトとみなさなくなる。

【0073】

機器Bは、 T_b を機器Aに送信後、図11のステップ1106～ステップ1108と同様の手順で、公開値Yと共有鍵Kを求める(ステップ108～ステップ110)。公開値Yの計算が終わると、機器Aに対して公開値Yを送る(ステッ

プ112)。

【0074】

機器Aでは、機器Bから公開値Yを受け取ると、応答待ち期限：T_bが満了していないため、D i f f i e - H e l l m a nの公開値の応答と判断して正しく処理できる。従って、機器A側でも共有鍵Kを計算し（ステップ113）、機器A、機器Bの両方で鍵の交換が成功する。

【0075】

以上のように、低い性能のCPUを搭載した機器Bは、鍵交換に伴う高負荷の演算の所要時間を事前に推定し、応答メッセージの最大遅延時間を対向の機器（機器A）に予告しておく。これにより、従来の方法で課題であった、低い性能のCPU搭載機器がまだ鍵交換の演算をしている間に、対向の機器がタイムアウトと判断してしまうような課題が発生せず、鍵交換が正常に成功できるようになる。

【0076】

なお、図1において機器Bは、相手（機器A）から公開値Xを受信した後に、演算所要時間T_bを予告するメッセージを送り返しているが、演算所要時間を相手に予告する手順は、図1の手順に限定されるものではない。

【0077】

例えば、機器AがD i f f i e - H e l l m a nの公開値Xを計算し始める前に、予め機器A、機器B間でお互いの演算所要時間を通知する手順を行っても良い。具体的には、図4のステップ401で、機器A側から所要時間の問い合わせ要求を発行し、それに対して機器Bが演算所要時間を推定、応答する（ステップ401～ステップ403）。ステップ402、ステップ403は、図1のステップ106、ステップ107と同様の手順である。その後の処理は、図1のステップ101以降の処理（ただしステップ106、107を除く）と同じである。

【0078】

また、実施の形態1では、D i f f i e - H e l l m a nの鍵交換時に使う演算として、離散対数問題を利用したべき乗／モジュロ演算のケースを説明したが、本発明の適用範囲はこの演算に限定されるものではない。例えば、D i f f i

e-Hellmanの変形として、楕円曲線に基づく数学的集合を用いた演算方法を使っても良い。

【0079】

また、Diffie-Hellmanに関する演算所要時間の推定方法（図1のステップ106、図4のステップ402の処理）として、事前に計算した結果を保存しておく方法を説明したが、この方法に限定するものではない。例えば、その時点のCPU使用率を測定し、CPU使用率が既に他のアプリケーション等によって使われている場合には、演算所要時間を長く推定してもよい。さらに、複数の機器との間の鍵交換が同時期に集中する可能性がある場合には、複数のDiffie-Hellmanの処理がキューに溜まる可能性がある。そのため、Diffie-Hellmanの処理待ちキューなどを設け、処理順番に応じて演算所要時間を長く推定しても良い。

【0080】

また、実施の形態1の処理シーケンスとして図1では、Diffie-Hellmanの演算と公開値のやりとり部分に限定して説明したが、その他のDiffie-Hellmanを利用する鍵交換プロトコルに適用してもよい。例えば、図12で説明したように、IKEのシーケンスでは2箇所でDiffie-Hellmanの鍵交換を利用しているが、実施の形態1で説明したメッセージを、IKEのその他のメッセージと連携させて実行してもよい。具体的には、図4のステップ401とステップ403で行っている演算所要時間の問い合わせ・通知は、IKEのPhase1の各種パラメータの折衝（図12のステップ1202）の中で行っても良い。

【0081】

また、実施の形態1では、機器Bにおける共有鍵Kの算出は、自身の公開値Yの算出と同時に行っていた（公開値Yの送信前に算出していた）が、必ずしも公開値Yの送信前に行う必要はない。機器A側での共有鍵Kの算出は、公開値Yの受信後に行うので、むしろ、機器B側でも公開値Yの送信後に共有鍵Kを算出した方が鍵交換全体の処理時間が短く抑えられる場合がある。なお、機器Bにおける共有鍵Kの算出が、公開値Yの送出後に行われる場合には、実施の形態1の処

理手順において、機器Bでの演算所要時間の推定値： T_b （図1のステップ106）は、公開値Yの算出のみの時間として推定する必要がある。

【0082】

なお、実施の形態1では、機器Bの演算所要時間だけを機器Aに通知・報告する手順を説明したが、機器Aと機器Bがお互いの演算所要時間を通知・報告し合っても良い。

【0083】

（実施の形態2）

実施の形態1では、公開値Yや共有鍵Kの演算の所要時間を問い合わせ、通知することにより、タイムアウトの発生を防止する方法であったが、所要時間を通知しない方法を、図5を用いて、次に説明する。

【0084】

図5において、ステップ501～ステップ504は、図1におけるステップ101～ステップ104と同様である。機器Aは、ステップ505において、公開値Xを機器Bに通知する。機器Bは、公開値Yの演算を依頼されたが、後で公開値Yを送信することのみを機器Aに通知し（ステップ506）、秘密情報bを選定し（ステップ507）、公開値Yの演算（ステップ508）と共有鍵Kの演算（ステップ511）を始める。ステップ506により、後で公開値Yが送信されることを通知された機器Aは、再送間隔T[秒]を設定し、再送間隔Tが経過すると、状況確認・再送をステップ509において行なう。

【0085】

なお、ステップ505、ステップ506において、パケットロスが起きることもあるので、ステップ505の後で、機器Aは一定の再送間隔を設定して、再送間隔以内に機器Bからの通知がない場合、ステップ509と同様の状況確認・再送を行なうようにしてもよい。

【0086】

機器Bは、公開値Y、共有鍵Kの演算が終了していないので、ステップ510において、再び、後で公開値Yを送信することのみを機器Aに通知する。機器Aは、再送間隔Tが経過するとステップ512において、状況確認・再送を行なう

。機器Bは、まだ、公開値Y、共有鍵Kの演算が終了していないので、ステップ513において、再び、後で公開値Yを送信することのみを機器Aに通知する。機器Aは、再送間隔Tが経過するとステップ514において、状況確認・再送を行なう。機器Bは、まだ、公開値Y、共有鍵Kの演算が終了していないので、ステップ515において、再び、後で公開値Yを送信することのみを機器Aに通知する。その後で、公開値Y、共有鍵Kの演算が終了し、公開値Yを機器Aに送信する（ステップ516）。機器Aは、ステップ517において、入手した公開値Yと秘密値aにより共有鍵Kを算出し、鍵交換が終了する。機器Aは、機器Bから、ステップ506、ステップ510、ステップ513、ステップ515により、後で公開値Yを送信することの通知を受けたら、図7で説明したタイムアウト時間をリセットし、新たにタイムアウト時間の計測を始める。再送間隔T [秒]をタイムアウト時間より短くしておけば、タイムアウトの発生はなくなる。

【0087】

また、機器Bが一旦、ステップ506のような “後で公開値Yを送信する旨の通知” を行くと、機器A側ではタイムアウト時間を設定せずに無限に待ちつづける処理に切り替えても良い。

【0088】

なお、ステップ516の公開値Yの通知は、ステップ508の後に行なってもよい。この場合は、機器Aは、公開値Yを受信した後は、機器Bに対して、共有鍵Kの演算が終了したかどうかの質問を、状況確認として行ない、これに対して、機器Bは、共有鍵Kの演算中であることを、機器Aに回答し、機器Aは、回答を受けた後、タイムアウト時間をリセットし、新たにタイムアウト時間の計測を始め、機器Bから共有鍵Kの演算終了の回答が帰ってくるまで、繰り返すようにすればよい。機器Aのステップ517が終了し、機器Bから共有鍵Kの演算終了の回答を得たら、共有鍵交換が完了する。

【0089】

現状のIKEの規定では、機器Bが公開値Yを通知したときに、共有鍵Kの算出まで完了しているべきか否かの規定はなく、機器A側で機器Bの共有鍵Kの算出完了を、タイムアウト時間を設けて待つような規定もない。従って、必ずしも

機器A側でのタイムアウト計測を行う必要はないのであるが、上記のように、機器A側でタイムアウト計測をしてもよい。

【0090】

(実施の形態3)

次に本発明の実施の形態3に係る共有鍵交換方法について説明する。実施の形態3は、従来の共有鍵交換方法における3つの課題のうち、第2の課題を解決するための方法である。

【0091】

まず、実施の形態3の概要について、図6を用いて説明する。先にも述べたように、同一に鍵を長時間にわたって使用すると、その間に鍵が第3者によって解読されることが全く無いとは言えない。そこで、鍵の作成に際して、その鍵に寿命を設定するようにし、鍵交換が成功するとその鍵には寿命が設けられているようにする。その鍵の寿命が満了になる前に、次の鍵の作成処理であるリキー処理を開始し、次の新しい鍵を作成し、使用中の鍵の寿命が満了となる前に、この新しい鍵に切り替える。実施の形態3では、リキー処理の完了時刻が、古い鍵の寿命満了時刻よりも前になるように、リキー処理の開始時刻を算出し、リキー処理を開始する方法である。

【0092】

図7は、本発明の共有鍵交換方式におけるリキー処理の手順を示す図である。機器Aは、ステップ701において、機器Bに、所要時間、すなわち、公開値Yの演算に必要な時間 T_{b_long1} と共有鍵Kの演算に必要な時間 T_{b_long2} を問い合わせる。機器Bは、ステップ703において、 T_{b_long1} と T_{b_long2} の推定を行ない、ステップ704において、機器Aに回答する。機器Aは、ステップ702において、機器Aが行なう公開値Xの演算に必要な時間 T_{a_long1} と共有鍵Kの演算に必要な時間 T_{a_long2} の推定を行ない、ステップ705において、 t_start を算定する。

【0093】

【数 7】

$$t_start < t_endA - (Ta_long1 + Ta_long2 \\ + Tb_long1 + Tb_long2) - \alpha$$

【0094】

α は、機器 A、機器 B 間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【0095】

ここで、 t_endA は、古い鍵の寿命満了時刻である。 Ta_long1 、 Ta_long2 、 Tb_long1 、 Tb_long2 は、それぞれ、演算の正味時間に、ステップ 706、ステップ 707 などのように、その前後に必要な付随する処理時間も含んだ値である。

【0096】

機器 A は、時刻 t_start が来ると、リキー処理を開始する。すなわち、ステップ 706、ステップ 707 において、変数 g と n を機器 A と機器 B の間で共有化し、ステップ 708 において、秘密値 a を生成し、ステップ 709 において、公開値 X を演算する。これらの処理に、 Ta_long1 の時間がかかる。機器 A は、ステップ 710 において、公開値 X を機器 B に通知する。機器 B は、ステップ 711 において、秘密値 b を生成し、ステップ 712 において、公開値 Y を演算し、ステップ 713 において、共有鍵 K を生成する。ステップ 711、ステップ 712 の処理に、 Tb_long1 がかかる。さらに、ステップ 713 の処理に、 Tb_long2 がかかる。機器 B は、ステップ 714 において、公開値 Y を機器 A に通知する。機器 A は、ステップ 715 において、共有鍵 K を演算して生成する。この処理に、 Ta_long2 がかかる。 t_start を（数 7）としたので、ステップ 715 の処理は、 t_endA までに完了する。すなわち、古い鍵の寿命満了以前に、新しい鍵が誕生した。以降は、この新しい鍵が使用される。

【0097】

また、ステップ 710 のあとに機器 A において、 $Tb_long1 + Tb_long2 + \alpha$ 以上のタイムアウト時間を設定することにより、第 1 の課題も解決

できる。

【0098】

ステップ714の、公開値Yの機器Bから機器Aへの通知は、ステップ712のあとですぐに行なってもよい。この場合は、 t_start を次の式により求める。

【0099】

【数8】

$$t_start < t_endA - \{Ta_long1 + Tb_long1 \\ + MAX(Ta_long2, Tb_long2)\} - \alpha$$

【0100】

α は、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【0101】

$MAX(Ta_long2, Tb_long2)$ は、 Ta_long2 と Tb_long2 のうち大きい方の値である。このようにすれば、 Ta_long2 、または、 Tb_long2 を長めに取りることが可能になる。

【0102】

ステップ701は、ステップ702、ステップ703、ステップ704、ステップ705に必要な時間 t_AB を見込んで早めに開始しなければならない。古い鍵を生成した直後に、ステップ701～ステップ705を実行して、次の t_start を算定しておけばよい。また、時刻($t_start - t_AB$)が来たら、ステップ701～ステップ705を再度実行して、その時の最新の所要時間を問い合わせ、 t_start を算定し直すようにしてもよい。

【0103】

処理時間の推定方法としては、古い鍵の交換時に行なった各ステップの処理時間、またはCPU負荷値を、図3のデータベース部303に機器A、機器Bでそれぞれ記憶しておき、その数値を元にしてもよい。また、ステップ701の問い合わせ時における機器A、機器Bの各CPU負荷値と、上記処理時間、またはCPU負荷値とから推定してもよい。

【0104】

ステップ704のあとに機器Bにおいて、 $Ta_long1 + \alpha$ 以上、ステップ710のあとに機器Aにおいて、 $Tb_long1 + \alpha$ 以上、ステップ714のあとに機器Bにおいて、 $Ta_long2 + \alpha$ 以上のタイムアウト時間を設定することにより、第1の課題も解決できる。

【0105】

図7では、機器Aから所要時間の問い合わせを行ない、 t_start の算定を行なうようにしたが、機器Aと機器Bを入れ替えて逆にしてもよい。

【0106】

なお、機器B側に、課題1のような問題がある場合は、ステップ704のあとに機器Bにおいて、 $Ta_long1 + \alpha$ 以上のタイムアウト値を設定してもよい。

【0107】

(実施の形態4)

次に本発明の実施の形態4に係る共有鍵交換方法について説明する。実施の形態4は、従来の共有鍵交換方法における3つの課題のうち、第2の課題を解決するための方法であって、図6において説明した、リキー処理の開始時刻を算出して、その時刻までにリキー処理を開始する方法である。本実施の形態では、それぞれの機器が自分の所用時間だけを基にして、リキー開始時刻を推定する。そして、CPU処理能力の低い方の機器が、結果的に、鍵更新のイニシエータになる。

【0108】

図8は、本発明の共有鍵交換方式におけるリキー処理の手順を示す図である。機器Aは、ステップ801において、機器Aが行なう公開値Xの演算に必要な時間 Ta_long1 と共有鍵Kの演算に必要な時間 Ta_long2 の推定を行ない、機器A自身のリキー開始時刻を次のように推定する。

【0109】

【数9】

$$t_startA < t_endA - (Ta_long1 + Ta_long2) \\ * 2 - \alpha$$

【0110】

ここで、 t_endA は、機器Aが使用中の共有鍵Kの寿命満了時刻、 t_startA は、機器Aのリキー開始時刻である。また、 α は、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【0111】

機器Bは、ステップ802において、 Tb_long1 と Tb_long2 の推定を行ない、機器B自身のリキー開始時刻を次のように推定する。

【0112】

【数10】

$$t_startB < t_endB - (Tb_long1 + Tb_long2) \\ * 2 - \alpha$$

【0113】

ここで、 t_endB は、機器Bが使用中の共有鍵Kの寿命満了時刻、 t_startB は、機器Bのリキー開始時刻である。また、 α は、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【0114】

Ta_long1 、 Ta_long2 、 Tb_long1 、 Tb_long2 は、それぞれ、演算の正味時間に、その前後に必要な付随する処理時間や揺らぎも含んだ値である。また、一般的に、 t_endA と t_endB は同じ値に設定されている。

【0115】

機器Bが機器Aよりも、CPU処理能力が低い場合、 t_startA より t_startB の方が小さい値となり、 t_startB が時間的に先になる。

【0116】

機器Bは、時刻 t_startB が来ると、リキー処理を開始する。すなわち、ステップ803において、変数 g と n を機器Aに通知し、リキー開始を宣言、

通知する。機器Aは、ステップ804において、変数gとnを受取ったことを応答、通知し、リキー開始を了解する。なお、新たな変数gとnを使わず、前の数値を使う場合は、リキー開始の通知とその了解の応答を行なうだけで良い。

【0117】

機器Bは、ステップ805において、秘密値bを生成し、ステップ806において、公開値Yを演算する。ステップ805、ステップ806の処理に、 T_{b_long1} かかる。機器Bは、ステップ807において、公開値Yを機器Aに通知する。機器Aは、自身のリキー開始時刻 $t_start A$ に関係なく、ステップ808において、秘密値aを生成し、ステップ809において、公開値Xを演算する。これらの処理に、 T_{a_long1} の時間がかかる。機器Aは、つぎに、ステップ810において、共有鍵Kを演算して生成する。この処理に、 T_{a_long2} かかる。機器Aは、ステップ811において、公開値Xを機器Bに通知する。機器Bは、ステップ812において、共有鍵Kを生成する。ステップ812の処理に、 T_{b_long2} かかる。

【0118】

機器Bは、 $t_start B$ を（数10）とした。すなわち、機器Bは、機器Aの行なうステップ805、ステップ806、ステップ807、ステップ812の処理についても、機器B自身の処理時間がかかると仮定したが、機器AのCPU処理能力は、機器Bのそれよりも高いので、ステップ812の処理は、 $t_end A$ 、 $t_end B$ までに完了することになる。すなわち、古い鍵の寿命満了以前に、新しい鍵が誕生した。以降は、この新しい鍵が使用される。

【0119】

機器A、機器BのうちCPU処理能力が低い方が、先にリキー処理の開始を宣言、通知するので、お互い相手の機器の公開値、共有鍵Kの演算所要時間を知る必要が無い。

【0120】

CPU処理能力の高い方の機器Aは、CPU処理能力の低い方の機器Bにしたがって鍵交換処理を開始し処理を進めるので、機器A自身のリキー開始時刻 $t_start A$ は、機器Aの各処理ステップの処理時刻とは関係なくなる。

【0121】

ステップ801、ステップ802は、早めに開始しなければならない。古い鍵を生成した直後に、ステップ801、ステップ802を実行して、次の $t_start A$ 、 $t_start B$ を算定しておけばよい。

【0122】

処理時間の推定方法としては、古い鍵の交換時に行なった各ステップの処理時間、またはCPU負荷値を、図3のデータベース部303に機器A、機器Bでそれぞれ記憶しておき、その数値を元にしてもよい。また、ステップ801、ステップ802の推測時における機器A、機器Bの各CPU負荷値と、上記処理時間、またはCPU負荷値とから推定してもよい。

【0123】

なお、第1の課題を解決するには、機器Bから $Tb_long1 + \alpha$ の時間を入手して、ステップ804のあとに機器Aにおいて、 $Tb_long1 + \alpha$ 以上のタイムアウト時間を設定すればよい。さらに、機器Bから $Tb_long2 + \alpha$ の時間を入手して、ステップ811のあとに機器Aにおいて、 $Tb_long2 + \alpha$ 以上のタイムアウト時間を設定してもよい。

【0124】

(実施の形態5)

次に本発明の実施の形態5に係る共有鍵交換方法について説明する。実施の形態5は、従来の共有鍵交換方法における3つの課題のうち、第2の課題を解決するための方法であって、図6において説明した、リキー処理の開始時刻を算出し、その時刻までにリキー処理を開始する方法である。本実施の形態では、それぞれの機器が自分の所用時間だけを基にして、リキー開始時刻を推定する。そして、CPU処理能力の低い方の機器が、結果的に、鍵更新のイニシエータになる。

【0125】

図9は、本発明の共有鍵交換方式におけるリキー処理の手順を示す図である。機器Aは、ステップ901において、機器Aが行なう公開値Xの演算に必要な時間 Ta_long1 と共有鍵Kの演算に必要な時間 Ta_long2 の推定を行ない、機器A自身のリキー開始時刻を次のように推定する。

【0126】

【数11】

$$t_startA < t_endA - (Ta_long1 + Ta_long2) - \alpha$$

【0127】

ここで、 t_endA は、機器Aが使用中の共有鍵Kの寿命満了時刻、 t_startA は、機器Aのリキー開始時刻である。また、 α は、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【0128】

機器Bは、ステップ902において、 Tb_long1 と Tb_long2 の推定を行ない、機器B自身のリキー開始時刻を次のように推定する。

【0129】

【数12】

$$t_startB < t_endB - (Tb_long1 + Tb_long2) - \alpha$$

【0130】

ここで、 t_endB は、機器Bが使用中の共有鍵Kの寿命満了時刻、 t_startB は、機器Bのリキー開始時刻である。また、 α は、機器A、機器B間のメッセージ送受信時の遅延時間と揺らぎを考慮した値である。

【0131】

Ta_long1 、 Ta_long2 、 Tb_long1 、 Tb_long2 は、それぞれ、演算の正味時間に、その前後に必要な付随する処理時間や揺らぎも含んだ値である。また、一般的に、 t_endA と t_endB は同じ値に設定されている。

【0132】

機器Bが機器Aよりも、CPU処理能力が低い場合、 t_startA より t_startB の方が小さい値となり、 t_startB が時間的に先になる。

【0133】

機器Bは、時刻 t_startB が来ると、リキー処理を開始する。すなわち、ステップ903において、変数 g と n を機器Aに通知し、リキー開始を宣言、通知する。機器Aは、ステップ904において、変数 g と n を受取ったことを応

答、通知し、リキー開始を了解する。なお、新たな変数 g と n を使わず、前の数値を使う場合は、リキー開始の宣言、通知とその了解の応答、通知を行なうだけで良い。

【0 1 3 4】

機器Bは、ステップ9 0 5において秘密値 b を生成し、ステップ9 0 6において公開値 Y を演算する。これらの処理に、 T_{b_long1} の時間がかかる。機器Bは、ステップ9 1 0において公開値 Y を機器Aに通知する。機器Aは、ステップ9 1 2において、共有鍵 K を演算して生成する。

【0 1 3 5】

機器Aは、ステップ9 0 7において秘密値 a を生成し、ステップ9 0 8において公開値 X を演算し、ステップ9 0 9において公開値 X を機器Bに通知する。ステップ9 0 7、ステップ9 0 8の処理に、 T_{a_long1} がかかる。機器Bは、ステップ9 1 1において、共有鍵 K を演算して生成する。この処理に、 T_{b_long2} がかかる。

【0 1 3 6】

機器Bより機器AのCPU処理能力が高い場合は、ステップ9 0 9がステップ9 1 0よりも先に生起する場合が多くなる。

【0 1 3 7】

機器Bは、 t_startB を（数12）とした。すなわち、機器Bは、機器Aの行なうステップ9 0 7、ステップ9 0 8、ステップ9 0 9、ステップ9 1 2の処理についても、機器B自身の処理時間がかかると仮定したが、機器AのCPU処理能力は、機器Bのそれよりも高いので、ステップ9 1 1、ステップ9 1 2の処理は、 t_endA 、 t_endB までに完了することになる。すなわち、古い鍵の寿命満了以前に、新しい鍵が誕生した。以降は、この新しい鍵が使用される。

【0 1 3 8】

機器A、機器BのうちCPU処理能力が低い方が、先にリキー処理の開始を宣言するので、お互い相手の機器の公開値、共有鍵 K の演算所要時間を知る必要が無い。

【0139】

CPU処理能力の高い方の機器Aは、CPU処理能力の低い方の機器Bにしたがって鍵交換処理を開始し処理を進めるので、機器A自身のリキー開始時刻 $t_start A$ は、機器Aの各処理ステップの時刻とは関係なくなる。

【0140】

ステップ901、ステップ902は、早めに開始しなければならない。古い鍵を生成した直後に、ステップ901、ステップ902を実行して、次の $t_start A$ 、 $t_start B$ を算定しておけばよい。

【0141】

処理時間の推定方法としては、古い鍵の交換時に行なった各ステップの処理時間、またはCPU負荷値を、図3のデータベース部303に機器A、機器Bでそれぞれ記憶しておき、その数値を元にしてもよい。また、ステップ901、ステップ902の推測時における機器A、機器Bの各CPU負荷値と、上記処理時間、またはCPU負荷値とから推定してもよい。

【0142】

なお、第1の課題を解決するには、機器Bから $Tb_long1 + \alpha$ の時間を入手して、ステップ904のあとに機器Aにおいて、 $Tb_long1 + \alpha$ 以上のタイムアウト時間を設定してもよい。さらに、機器Bから $Tb_long2 + \alpha$ の時間を入手して、ステップ909のあとに機器Aにおいて、 $Tb_long2 + \alpha$ 以上のタイムアウト時間を設定してもよい。

【0143】

(実施の形態6)

次に本発明の実施の形態6に係る共有鍵交換方法について説明する。実施の形態6は、従来の共有鍵交換方法における3つの課題のうち、第3の課題を解決するための方法である。

【0144】

まず、実施の形態6の概要について説明する。ネット家電のように性能の低いCPUを搭載している機器で、Diffie-Hellmanのような高負荷の処理を実行すると、CPUが占有されてしまい、長い処理期間中、他のアプリケ

ーションが正しく動作しなくなる可能性がある。

【0145】

一方、共有鍵交換は、常に最短時間で完了させる必要はない。具体的には、パケット通信を行う端末の要請により、鍵を新規に交換する場合には、鍵交換の遅延時間が端末のアプリケーションでの遅延としてユーザに見えてしまうので、できるだけ早く鍵交換を完了させる必要がある。しかしながら、一旦鍵を交換してしまった後、定期的に鍵を更新する場合は、ユーザからのパケットを古い鍵で暗号化処理するのと並行に、新しい鍵の交換を行うので、鍵交換自体の処理時間は長くてもよい。すなわち、古い鍵の寿命が切れる前に更新用の鍵交換が完了すれば良い。

【0146】

以上のことから、本発明の実施の形態6では、鍵交換の処理時間を最短で完了させるべきか否かを判定し、最短の処理が必要とされない状況の場合は、高負荷の処理を小単位に分けて実行し、時間的に負荷分散する。この処理により、鍵交換に伴う処理がCPUを占有せず、他のアプリケーションの動作を妨げないようにできる。また、対向の機器への応答として公開値を送信するまでの遅延時間が長くなるが、鍵の更新のように鍵交換の処理遅延の長さがユーザのパケット送受信に影響を与えない状況なので、問題にならない。

【0147】

以下、図10を使って、本発明の実施の形態6に係る共有鍵交換方法の概要を説明する。図10において、機器Aと機器Bは、暗号・認証処理と共有鍵交換を終端する機器の組とする。具体的には、図2中の機器間で共有鍵交換の処理を終端する3つの組み合わせ、すなわち(1)GW型機器A201とGW型機器B202、(2)GW型機器A201とホスト型機器B211、(3)ホスト型機器A210とホスト型機器B211のいずれかの組み合わせに対応するものとする。また、図10中の機器Bは、機器Aに比べて低い性能のCPUを搭載した機器とする。また、図10の機器Aまたは機器Bに関する処理シーケンスは、図3における共有鍵交換部301で行う処理に対応するものとする。

【0148】

まず、機器Bでは事前に、D i f f i e - H e l l m a n 交換の公開値 Y と共有鍵 K の演算に伴う全処理を、複数の均等な処理単位（以下、小処理単位と呼ぶ）に分けておく。さらに、単位時間の間に小処理単位を 1 単位分実行したときの C P U 使用率：U c p u を実測した結果を記録しておくものとする。ここで、機器Bが行うべき全演算に対する小処理単位の数を、T o t a l とする。

【0149】

また、機器Aと機器Bの間では、既に共有鍵 K__o l d を共有しており、その鍵の寿命が未来の時刻：t__e n d に切れる（使用不可の状態になる）ものとする。

【0150】

機器Aは、共有鍵 K__o l d の鍵交換中または交換後に、機器Bに対して、鍵更新時の所要時間を問い合わせる（ステップ1001）。機器Bでは、その時点の他のアプリケーションの平均的な C P U 使用率を測定し、他のアプリケーションの処理を維持したまま、D i f f i e - H e l l m a n の演算にどれだけの C P U 使用率を使えるか、単位時間に実行できる小処理単位の数はいくらかを推定する。具体的には、以下の式により、機器Bから機器Aに応答を返すまでの遅延時間を求める（ステップ1002）。

【0151】

【数13】

D c p u : 残り C P U 使用率

= 100 - 他のアプリケーションの C P U 使用率平均

【0152】

【数14】

単位時間の間に実行可能な D i f f i e - H e l l m a n の処理単位数

= D c p u / U c p u

【0153】

【数15】

遅延時間: Tb_long

$$= Total \times Ucpu / Dcpu + \alpha$$

ただし、 α : 固定値

【0154】

機器Bは、推定した遅延時間: Tb_long を、機器Aに送信する（ステップ1003）。

【0155】

一方、機器A側でも自分のDiffie-Hellmanに伴う遅延時間: Ta_long を推定しておき、共有鍵 K_old を更新するために鍵交換を開始すべき時刻: t_start を求める（ステップ1004）。

【0156】

【数16】

$$t_start = t_end - (Ta_long + Tb_long)$$

【0157】

機器Aでは、共有鍵 K_old の更新開始タイミングとして、時刻: t_start が到来していないかチェックし、 t_start の時刻（ステップ1005）になると、以下の手順で鍵交換を実行する。

【0158】

機器Aは、Diffie-Hellmanの公開値Xを計算し、機器Bに送信する（ステップ1006～1008）。機器Bは公開値Xを受信すると、その要求が既に存在する鍵の更新用なのか、新規に鍵を作成するための要求かを判定する（ステップ1009）。具体的には、鍵の情報を記憶させているデータベース（図3中のデータベース部に対応）を検索し、対応する鍵が存在する場合は「鍵の更新」、存在しない場合は「新規作成」と判定する。機器Bは、判定結果に応じて以後のDiffie-Hellmanの演算の実行方法を変える。すなわち、鍵の「新規作成」と判定された場合には、機器Bとしての最短の時間で実行し、「鍵の更新」と判定された場合には、機器Aに通知した Tb_long の時間をかけてゆっくり処理する。図10のシーケンスでは、 K_old が既に存在す

るので、「鍵の更新」と判定される。機器Bは単位時間あたりに、 $Dcpu/Ucpu$ 個の小処理単位を実行し、 Tb_long の時間をかけて公開値Yを算出し、機器Aに送信する（ステップ1010～1013）。

【0159】

一方、機器Aも、少なくとも機器Bから通知された Tb_long の時間だけ、機器Bからの応答を待ち、公開値Yを受信した後、新しい共有鍵 K_new を算出する（ステップ1014）。

【0160】

以上のように、本発明の実施の形態6では、鍵の更新のような鍵交換の処理遅延時間が問題にならないケースに限定して、鍵交換に伴う高負荷の演算を時間的に負荷分散する。これにより、低性能のCPUを搭載した機器でも、鍵交換の処理が長期間にわたってCPUを占有することがなくなり、同じCPU上で動作する他のアプリケーションも正しく動作できるようになる。さらに、高負荷の演算を時間的に負荷分散する際にも、鍵の寿命（使用不可になるタイミング）を考慮することで、鍵交換の処理遅延時間が通常より長くなっても、暗号・認証処理を問題なく継続できる。なお、一例として、機器Bにおいて、 $Diffie-Hellman$ の演算に必要なCPUの処理量が $Im=200$ （ MI =メガインストラクション）であり、複数の均等な処理単位、すなわち、小処理単位として2（ MI ）に分ける場合、 $Total$ の値は、100（処理単位）である。機器BのCPUの処理能力が100（ $MIPS$ =メガインストラクション/秒）の場合、小処理単位の2（ MI ）を1単位分だけ、単位時間の間に実行するときのCPU使用率 $Ucpu$ は、2%である。機器BのCPUの処理能力を他のアプリケーションに50 $MIPS$ だけ使用する場合、残りCPU使用率 $Dcpu$ は、50%である。従って、単位時間に実行可能な $Diffie-Hellman$ の処理単位数 $Dcpu/Ucpu$ は、25（処理単位/秒）である。よって、 $Total \times Ucpu/Dcpu$ は、 $100/25=4$ 秒になる。20%の揺らぎなどの余裕を見込めば、遅延時間は、4.8秒とすればよい。

【0161】

（実施の形態7）

図10の実施の形態6において、処理の不可分散を良く行える方法を次に説明する。このために、実施の形態7では、鍵の交換が終了したら、機器A、Bは、可能なかぎり速やかに共有鍵の更新処理に入る。まず、機器Bは、ステップ1002においてCPUの使用量 I_m の推定を行なう。CPUの使用量 I_m は、例えば、機器Bが実施した鍵の作成処理の実行命令数（単位：MI：メガインストラクション）で表すことができる。前回のCPU使用量 I_m を記憶しておいてその値を使用してもよい。機器Aは、ステップ1004において、自身の秘密値 a の生成、公開値 X の演算、共有鍵 K の演算に必要な時間の合計値（ $T_{a_long1} + T_{a_long2}$ ）を（鍵の寿命完了時刻－現在時刻）から差し引いた機器B用割り当て時間 TW_b を求め、ステップ1008において、公開値 X と共に、機器Bに通知する。機器Bは、ステップ1009において、単位時間当たりのCPU使用率（ $I_m / TW_b + \beta$ ）をCPUに割り当てて、ステップ1010の秘密値 b の生成、ステップ1011の公開値 Y の演算、ステップ1012の共有鍵 K の演算を行なう。 β は、処理の揺らぎや前後に必要な処理を行なうための余裕分である。一般的に、 TW_b を大きくすれば、単位時間当たりのCPU使用率（ $I_m / TW_b + \beta$ ）を、100%より十分小さい値にすることができる。そうすれば、残ったCPU使用率の部分を、鍵交換処理以外のアプリケーションに振り向ける余裕ができる。 T_{b_long} の代りに、機器B用割り当て時間 $TW_b + \alpha$ が、タイムアウト待ち時間として設定される。図10において、ステップ1001は不要になる。

【0162】

このようにすれば、機器Bにおいて、鍵交換処理に割く単位時間当たりのCPU処理量を常に低くできるので、他の処理に回せるCPU処理量を多めに確保しておける。また、上記単位時間当たりのCPU使用率（ $I_m / TW_b + \beta$ ）を割り当ての最小量とし、他の処理の生起が少なくてCPU処理能力に余裕がある時間帯には、より大きいCPU使用率を割り当てて、早めに鍵交換処理を進めるようにしてもよい。

【0163】

なお、本実施の形態と同様の考え方を、図8の（実施の形態4）の演算におい

ても、適用できる。

【0164】

(実施の形態8)

図9の実施の形態5において、処理の負荷分散をよく行なえる方法を次に説明する。このために、実施の形態8では、鍵の交換が終了したら、機器A、Bは、できる限り速やかに鍵の更新処理に入る。まず、機器Bは、ステップ902においてCPUの使用量 I_m の推定を行なう。前回のCPU使用量 I_m を記憶しておいてその値を使用してもよい。機器Bは、余裕時間 $TY_b = (\text{鍵の寿命満了時刻} - \text{現在時刻})$ を算出し、単位時間当たりのCPU使用率 $= (I_m / TY_b + \beta)$ をCPUに割り当てて、ステップ905の秘密値 b の生成、ステップ906の公開値 Y の演算、ステップ911の共有鍵 K の演算を行なう。 β は、処理の揺らぎや各演算の前後に必要な処理を行なうための余裕分である。一般的に、 TY_b を大きくすれば、単位時間当たりのCPU使用率 $(I_m / TY_b + \beta)$ を、100%より十分小さい値にすることができる。そうすれば、残ったCPU使用率の部分を、鍵交換処理以外のアプリケーションに振り向ける余裕ができる。

【0165】

一方、機器Aは、ステップ901において自身のCPU使用量の推定を行なう。前回のCPU使用量を記憶しておいてその値を使用してもよい。前回のCPU使用量は、機器Aにおいても、 I_m である。機器Aは、余裕時間 $TY_a = (\text{鍵の寿命満了時刻} - \text{現在時刻})$ を算出し、単位時間当たりのCPU使用率 $= (I_m / TY_a + \beta)$ をCPUに割り当てて、ステップ907の秘密値 a の生成、ステップ908の公開値 X の演算、ステップ912の共有鍵 K の演算を行なう。なお、機器AにおけるCPU使用量が、機器Bの場合とは異なる場合には、機器AにおけるCPU使用量を用いればよい。

【0166】

なお、ステップ903、ステップ904のリキー開始の宣言は、なくともよい。また、機器A、Bの演算進行速度が、ほぼ同じになるので、ステップ909の公開値 X の通知、ステップ910の公開値 Y の通知は、ほぼ同じ時刻に行われることになり、機器BのCPU処理能力が低くとも、ステップ911の処理の完了

が、 $t_end B$ に間に合わなくなることはない。

【0167】

このようにすれば、機器Bにおいて、鍵交換処理に割く単位時間当たりのCPU処理量を常に低くできるので、他の処理に回せるCPU処理量を多めに確保しておける。また、上記単位時間当たりのCPU使用率 $(I_m / T Y b + \beta)$ を割り当ての最小量とし、他の処理の生起が少なくてCPU処理能力に余裕がある時間帯には、より大きいCPU使用率を割り当てて、早めに鍵交換処理を進めるようにしてもよい。

【0168】

(実施の形態9)

上記、各実施の形態において、第1の通信機器、第2の通信機器が、共有鍵Kの演算を終了したら、相手の機器に対して、共有鍵Kの生成完了の通知を行なうようにしてもよい。この場合、この通知を受信するまで、タイムアウトの検出を停止するようにしてもよい。そうすれば、第1の課題を解決できる。

【0169】

(実施の形態10)

先に、鍵の寿命を設定できることを説明した。この設定は、一般的に、図13の各種パラメータの折衝ステップ1202において行われる。この寿命を、 $(T a_long 1 + T a_long 2)$ と $(T b_long 1 + T b_long 2)$ を基準にして、十分余裕をもった値に設定することにより、鍵の寿命に到達する前に、リキーを完了できるようにしてもよい。鍵交換の完了時間が、 $(T a_long 1 + T a_long 2)$ と $(T b_long 1 + T b_long 2)$ の和によって決まる、図7の(実施の形態3)、図8の(実施の形態4)の場合は、 $(T a_long 1 + T a_long 2) + (T b_long 1 + T b_long 2)$ により、図9の(実施の形態5)の場合は、 $(T a_long 1 + T a_long 2)$ と $(T b_long 1 + T b_long 2)$ のうち、大きい方の値を基準にすればよい。例えば、機器BのCPU処理能力が大きくなり、 $(T b_long 1 + T b_long 2)$ が長い場合、CPU処理能力が低いほど、鍵の寿命を長くすればよい。

【0170】

上記基準の時間の数倍、あるいは10倍程度の寿命を設定すれば、CPU処理能力が低くとも、鍵交換処理以外の処理に振り向けるCPU処理能力の余裕が生れる。

【0171】

なお、 $(Ta_long1 + Ta_long2)$ と $(Tb_long1 + Tb_long2)$ としては、最初の鍵交換処理の際にかかった処理時間を記憶しておき、その値を使用してもよい。

【0172】

図1の実施の形態1においては、機器Bは、ステップ107において、応答の予定時間遅れを通知し、機器Aは、その時間だけ、タイムアウトを遅らせるようにした。また、図5の実施の形態2においては、状況確認・再送に対して公開値の送信が、遅れることを、機器Bが機器Aに通知し、機器Aがタイムアウトを発生させないようにした。このように、本発明では、応答が遅れることを応答予告として相手に通知することにより、相手側では、種々の対応策を採用することができるようになり、タイムアウトなどによる破綻を防止することを可能にする。

【0173】

上記、図5の実施の形態2において、課題1の対策、および、通信のパケットロスが発生する場合の対策として、状況確認・再送を行なったが、他の実施の形態においても、パケットロスの発生によるタイムアウトを避けるために、状況確認・再送を随時行ない、タイムアウト時間を更新するようにしてもよい。

【0174】

図8の実施の形態4、図9の実施の形態5において、 g 、 n の通知と了解の後に、秘密値 a 、または、 b の生成を行なうようにしたが、秘密値と公開値の生成を行なっておき、 g 、 n 、公開値 X 、または、 Y を、同時に通知するようにしてもよい。

【0175】

上記各実施の形態においては、図9の実施の形態5を除いて、最初に、 g 、 n を通知したり、パラメータの折衝を開始する側の機器をイニシエータとし、イニ

シェータの方が、秘密値の生成、公開値の演算により得た公開値を、先に相手方に送る例を、主として説明した。反対に、イニシェータでない方の機器が、先に公開値を算出して送る場合でも、本発明を適用することが可能である。

【0176】

上記、図7、図8、図9で説明した、鍵の寿命満了までにリキーを行なう各実施の形態において、 $t_end A$ 、 $t_end B$ は、ほぼ同じ時刻になるとして説明したが、鍵の寿命は、機器A、機器Bにおいて、それぞれの共有鍵の生成完了時点から起算して設定されるように規定されている場合がある。一般的に、機器A、機器Bの共有鍵の生成完了時点は、共有鍵の生成手順と、機器A、機器BのCPU処理能力などによって左右され、同時刻となることは希であり、 $t_end A$ と $t_end B$ の時刻には、 $Ta_long 2$ 、あるいは、 $Tb_long 2$ 程度の差がでる。

【0177】

図7の場合は、 $t_end A$ は、 $t_end B$ より($Ta_long 2$)だけ遅くなる。(ステップ712)の直後にステップ714を実行する場合は、 $t_end B$ と $t_end A$ の時刻の差、($t_end B - t_end A$)は、($Tb_long 2 - Ta_long 2$)になる。

【0178】

図8の場合は、 $T_end A$ は、 $T_end B$ より($Tb_long II$)だけ早くなる。ステップ809の直後にステップ811を実行する場合は、 $t_end B$ と $t_end A$ の時刻の差、($t_end B - t_end A$)は、($Tb_long 2 - Ta_long 2$)になる。

【0179】

図9の場合は、ステップ911とステップ912は、 $Ta_long 1$ と $Tb_long 1$ の遅い方が完了した後、同時に処理を開始することになるので、 $t_end B$ と $t_end A$ の時刻の差、($t_end B - t_end A$)は、($Tb_long 2 - Ta_long 2$)になる。

【0180】

したがって、(数7)～(数12)における $t_end A$ 、 $t_end B$ の値

は、上記 ($t_end B - t_end A$) の絶対値に相当する値に所定の余裕を加えた値だけ、時刻の早い数値を採用して計算するのが安全である。一般的に、秘密値、公開値、共有鍵の演算に要する時間の数倍から10倍以上の時間を鍵の寿命とする場合が多いので、 $t_end A$ 、 $t_end B$ の値を $Ta_long 2$ 、あるいは、 $Tb_long 2$ 程度早めの値とすることは、十分に可能である。なお、(数7)～(数12)をそのまま使用するには、鍵の満了時間が、設定した数値よりも上記 ($t_end B - t_end A$) に相当する値程度だけ短いものとして、 $t_end A$ 、 $t_end B$ の時刻を設定するようにすればよい。

【0181】

なお、鍵の寿命は、1) 鍵を生成した時刻から起算する場合、2) 鍵を使用して暗号通信を始めた時点から起算する場合、3) 暗号通信において一定数のパケットを処理するまで、4) 一定のバイト数を通信処理するまで、などの中から選択できるようになっている場合がある。2) 鍵を使用して暗号通信を始めた時点から起算する場合、3) 暗号通信において一定数のパケットを処理するまで、4) 一定のバイト数を通信処理するまで、を選択した場合は、機器Aと機器Bのそれぞれの鍵の寿命満了時刻は、実質的に、同一時刻となる。したがって、上述した、鍵の寿命満了時刻を早めとする配慮は無くともよく、適当な余裕値分だけ早めとすれば十分である。

【0182】

ステップ107、ステップ403で説明した時間 Tb 以内の応答を予告する応答予告のメッセージは、応答の遅延時間、応答の送信予定時刻などを送信すればよい。受信した機器Aは、応答予告の時間または時刻を基に、タイムアウトの時間設定を行えばよい。

【0183】

本発明は、暗号・認証処理を施したデータを送受信する2つの通信機器の間で共有鍵を交換する共有鍵交換方法であって、前記2つの通信機器は、第1の通信機器と第2の通信機器から成り、前記第1の通信機器および前記第2の通信機器は、それぞれ相手の機器から受信した公開値と自分が生成した秘密情報を基に、共有鍵を計算することができるものであれば、Diffie-Hellmanの

方法や I K E の方法以外の共有鍵交換方法に対しても適用可能である。

【0184】

なお、本発明の共有鍵交換方法のプログラムを記録した記録媒体は、プログラムを記録した R O M、R A M、フレキシブルディスク、C D - R O M、D V D、メモリカード、ハードディスクなどの記録媒体をいう。また、電話回線、搬送路などの通信媒体も含む概念である。

【0185】

【発明の効果】

以上のように、本発明によれば、実施の形態 1、実施の形態 2、実施の形態 6 のようにすることにより、対向の機器でのタイムアウト検出により、鍵交換が失敗するという、第 1 の課題が解決される。また、本発明によれば、実施の形態 3、実施の形態 4、実施の形態 5、実施の形態 6、実施の形態 7、実施の形態 8、実施の形態 10 のようにすることにより、鍵の寿命が設定されている場合に、寿命満了までに、次の鍵交換が完了しないという第 2 の課題が解決される。

【0186】

また、本発明によれば、実施の形態 6、実施の形態 7、実施の形態 8、実施の形態 10 のようにすることにより、共有鍵交換の処理が、機器上の他のアプリケーションの実行を妨害するという第 3 の課題が解決される。

【0187】

よって、第 2 の通信機器の搭載する C P U の性能が低い場合にも、負荷の高い鍵交換処理を正しく実行でき、共有鍵交換を成功させることができる。

【図面の簡単な説明】

【図 1】

本発明の実施の形態 1 における共有鍵交換方法の処理手順を示す図

【図 2】

本発明の共有鍵交換方法が適用されるネットワークの全体構成を示す図

【図 3】

本発明の共有鍵交換方法が適用される機器の機能的構成を示すブロック図

【図 4】

本発明の実施の形態 1 における共有鍵交換方法の別の処理手順を示す図

【図 5】

本発明の実施の形態 2 における共有鍵交換方法の処理手順を示す図

【図 6】

本発明の実施の形態 3 における鍵更新のタイムチャート

【図 7】

本発明の実施の形態 3 における共有鍵交換方法の処理手順を示す図

【図 8】

本発明の実施の形態 4 における共有鍵交換方法の処理手順を示す図

【図 9】

本発明の実施の形態 5 における共有鍵交換方法の処理手順を示す図

【図 1 0】

本発明の実施の形態 6 における共有鍵交換方法の処理手順を示す図

【図 1 1】

従来の共有鍵交換方法である D i f f i e - H e l l m a n の基本的な処理手順を示す図

【図 1 2】

R F C 2 4 0 7 ～ 2 4 0 9 で開示されている I K E の処理手順の概要を示す図

【図 1 3】

従来の共有鍵交換方法の問題点を説明するための図

【符号の説明】

2 0 1 G W 型機器 A

2 0 2 G W 型機器 B

2 0 3, 2 0 4, 2 0 5, 2 0 6 L A N 内端末

2 0 7 公衆網

2 0 8, 2 0 9 L A N

2 1 0 ホスト型機器 A

2 1 1 ホスト型機器 B

3 0 1 共有鍵交換部

3 0 2 データベース管理部

3 0 3 データベース部

3 0 4 暗号化／認証処理部

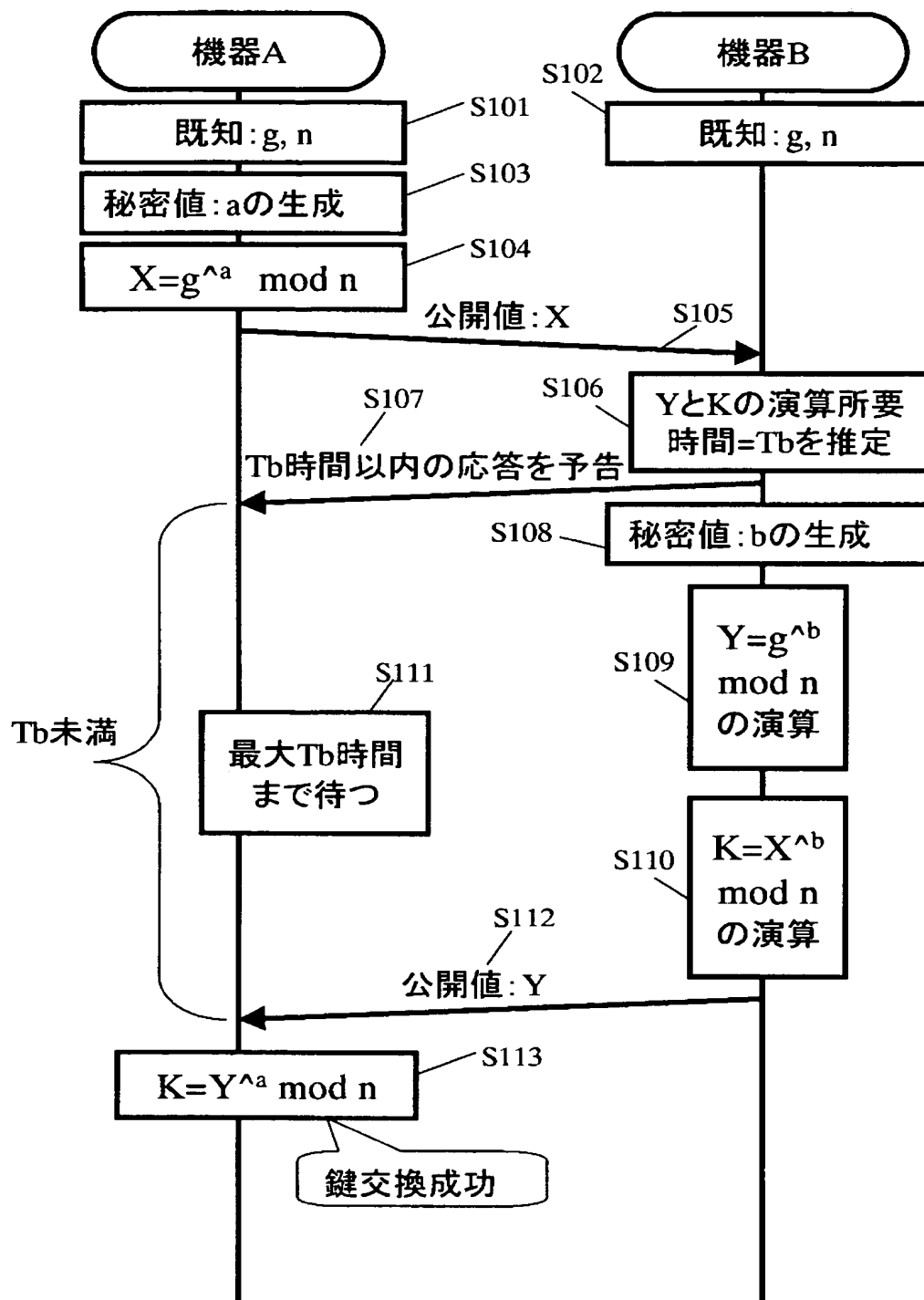
3 0 5, 3 0 6 通信プロトコル処理部

3 0 7 LAN I／F

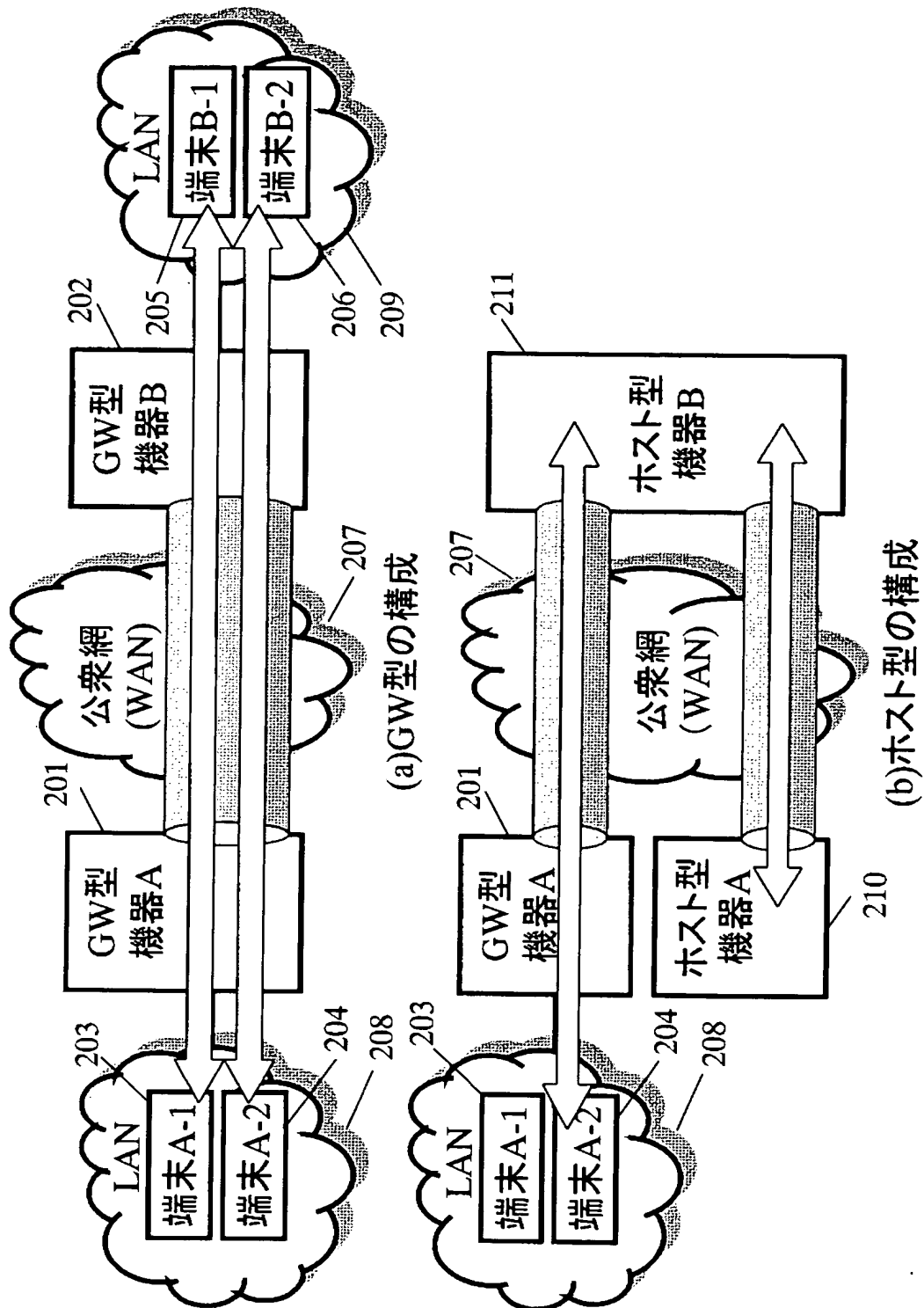
3 0 8 WAN I／F

【書類名】 図面

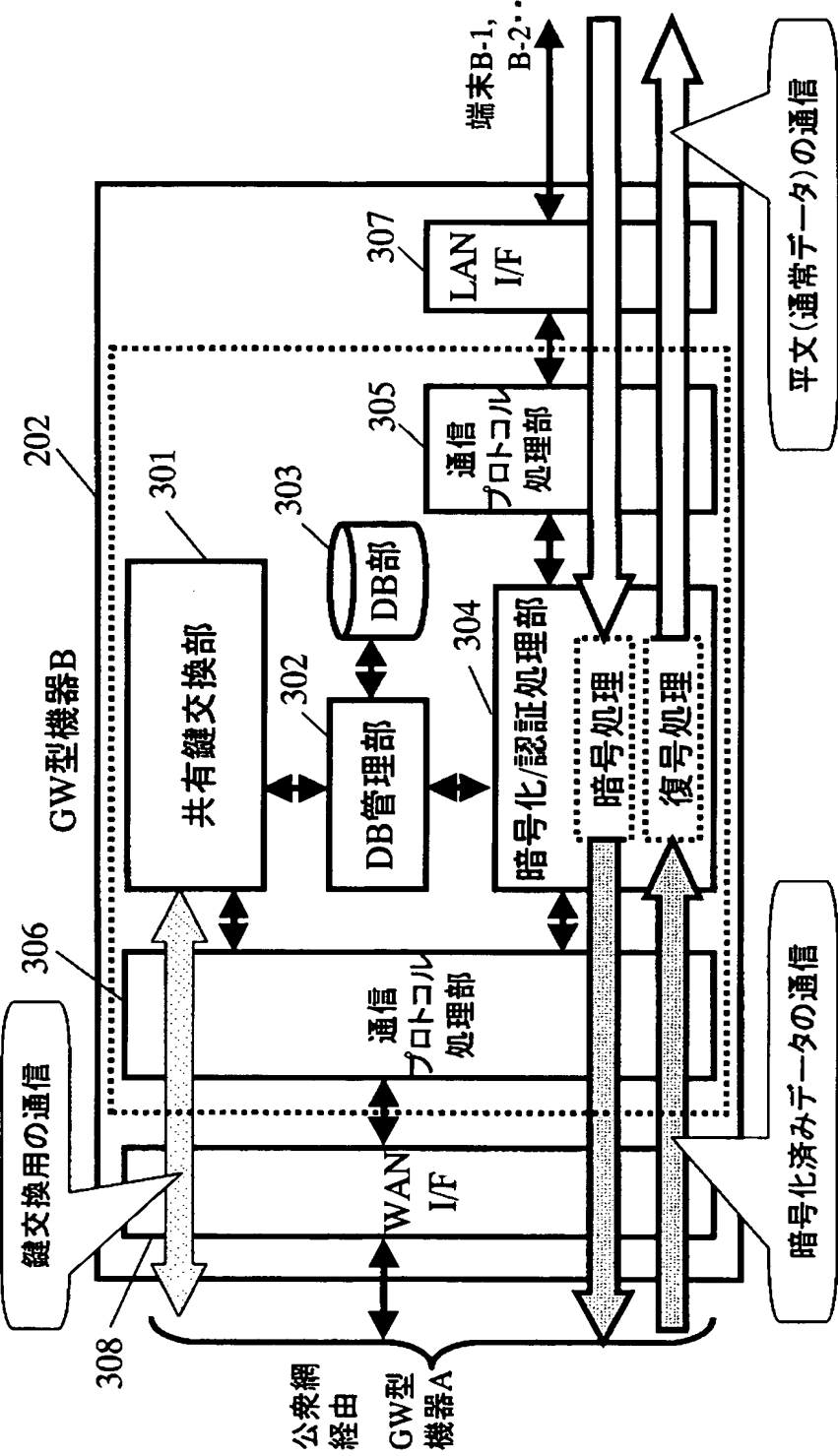
【図1】



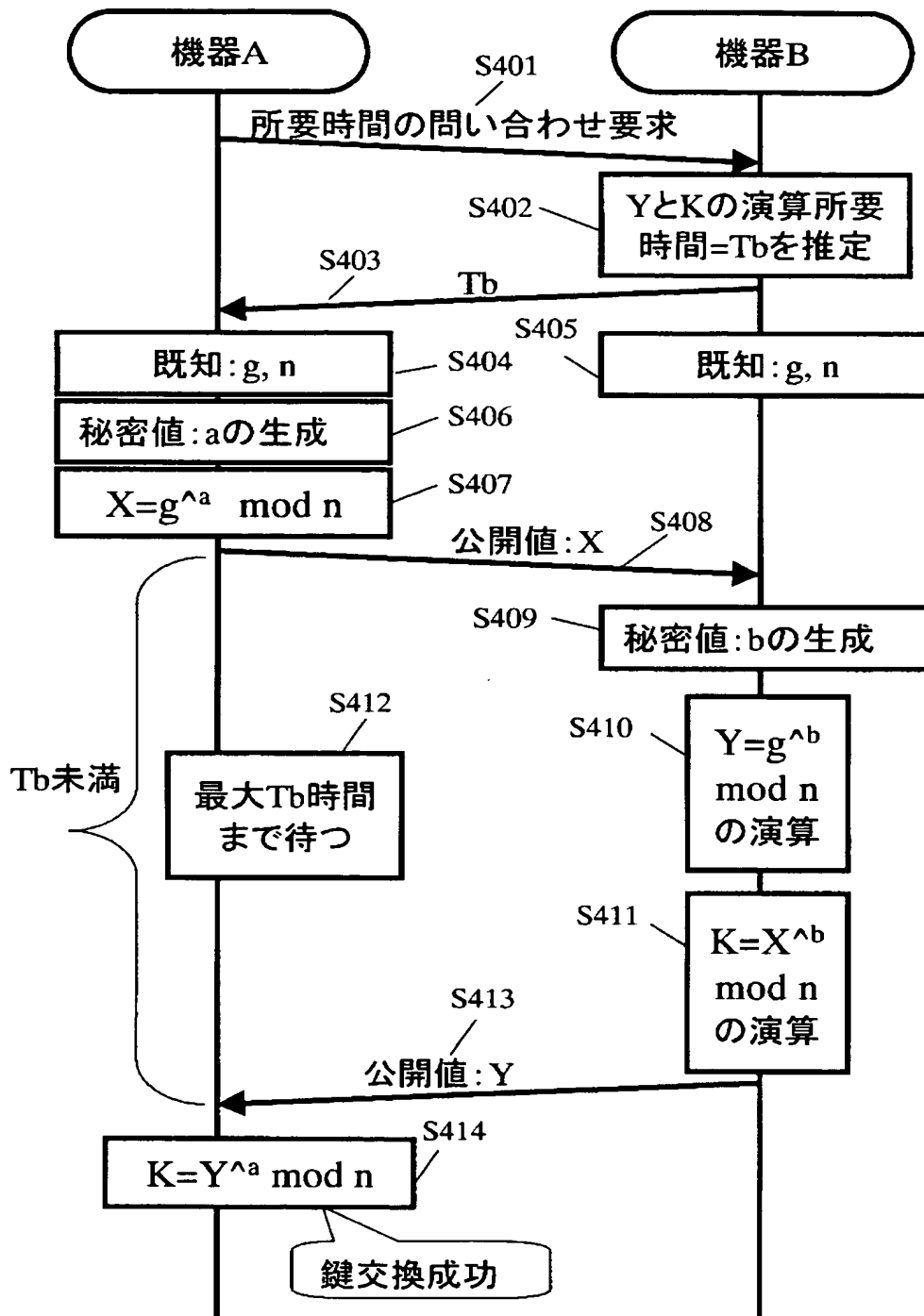
【図 2】



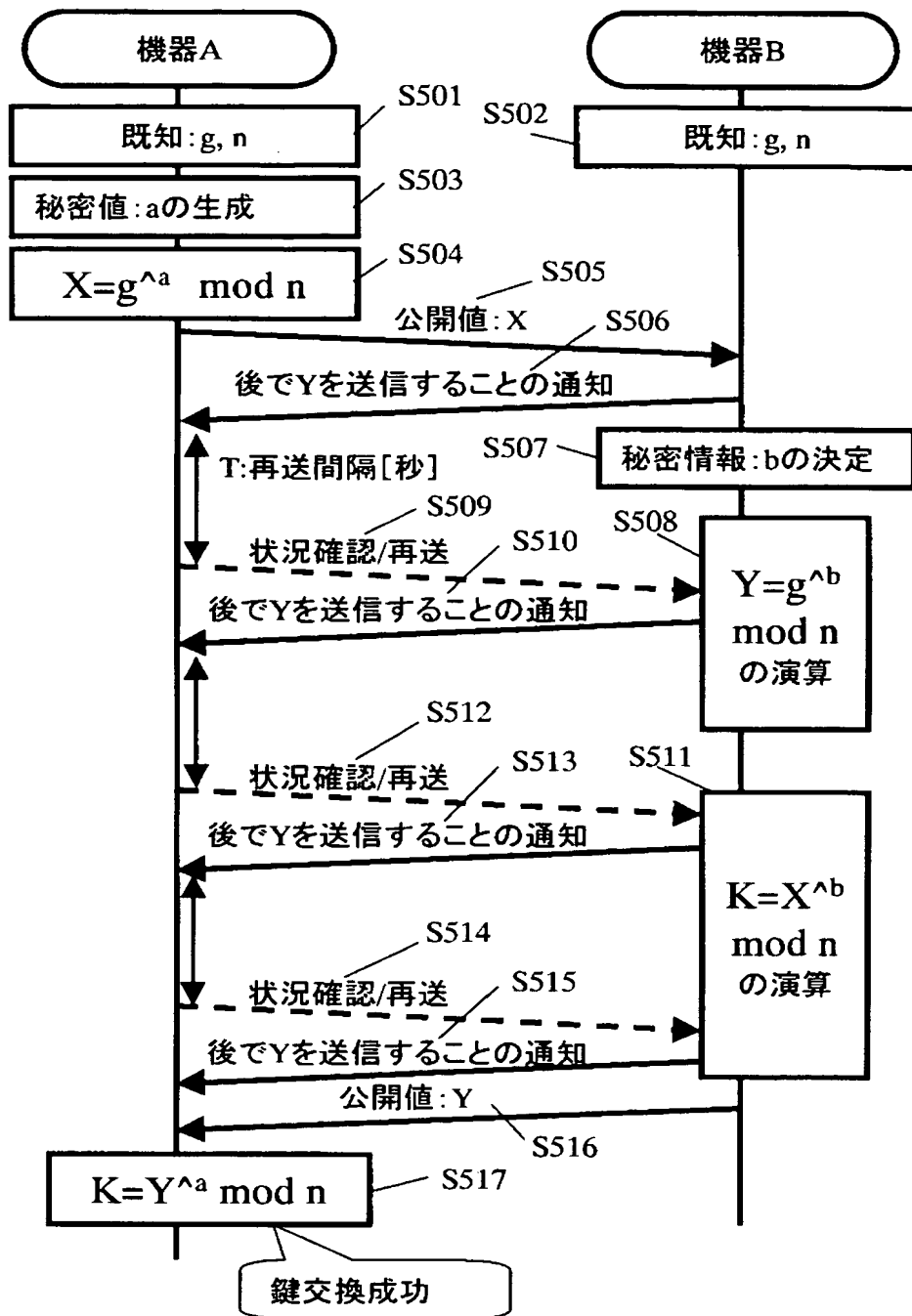
【図 3】



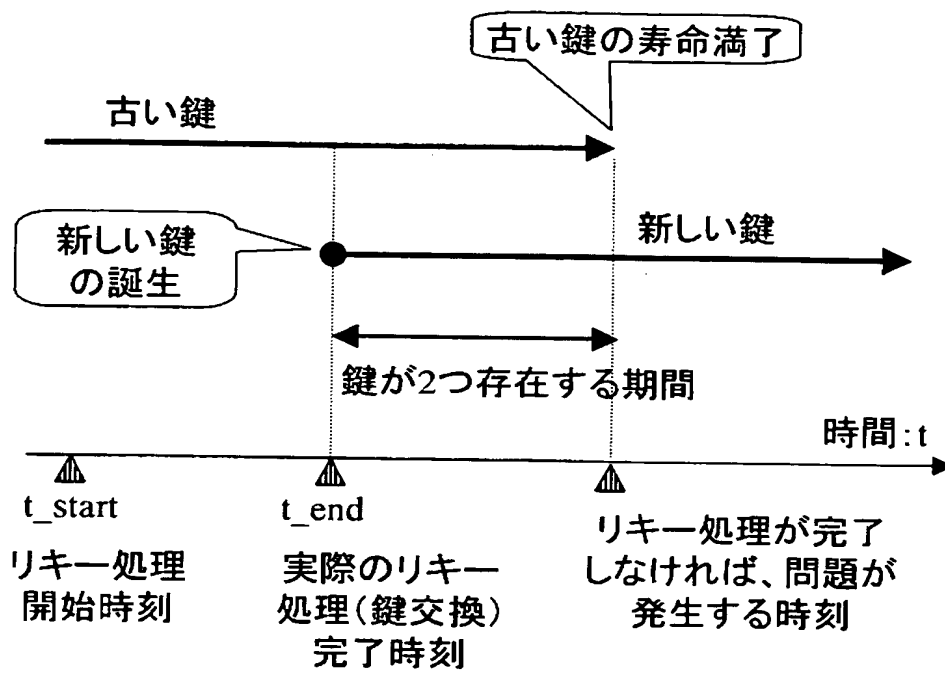
【図 4】



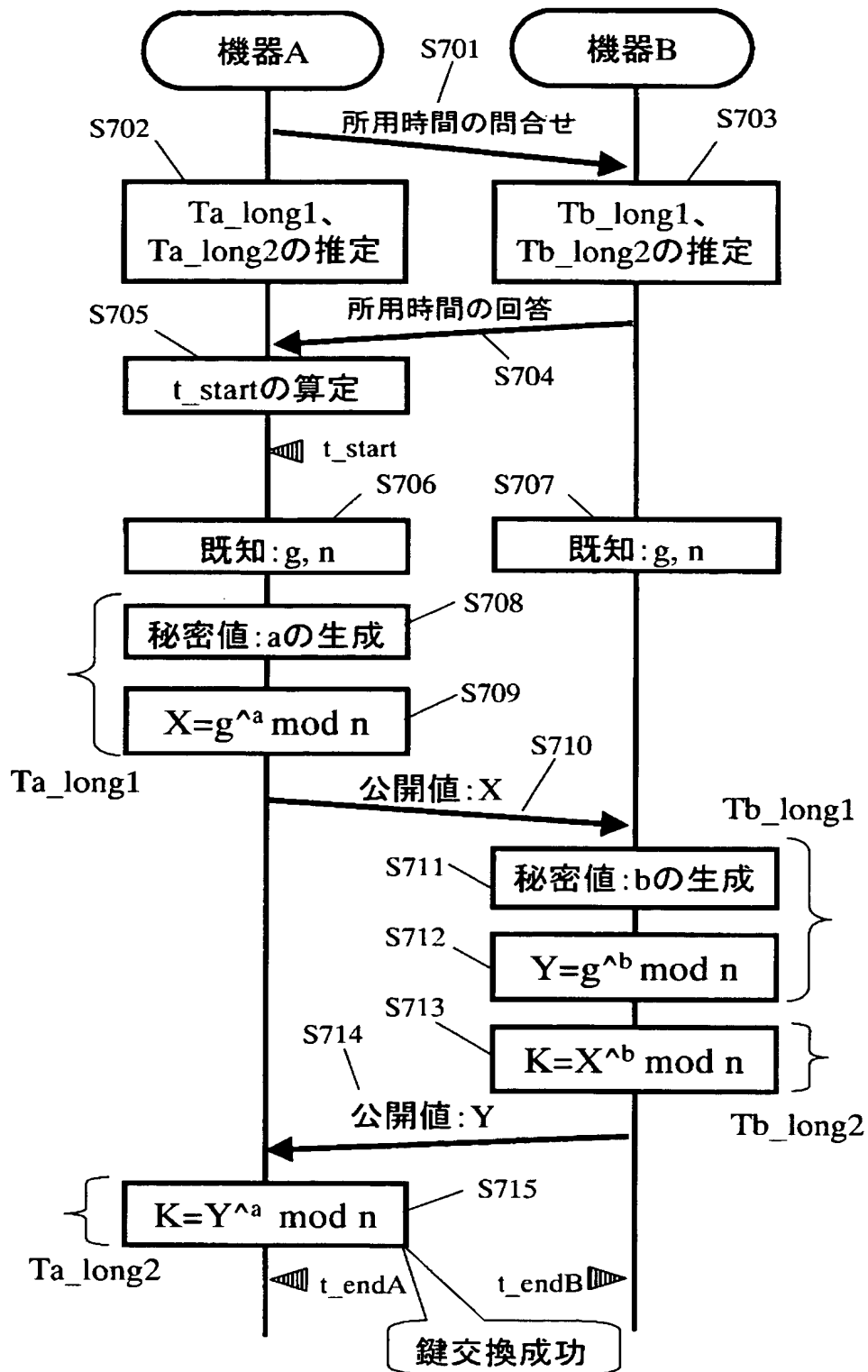
【図 5】



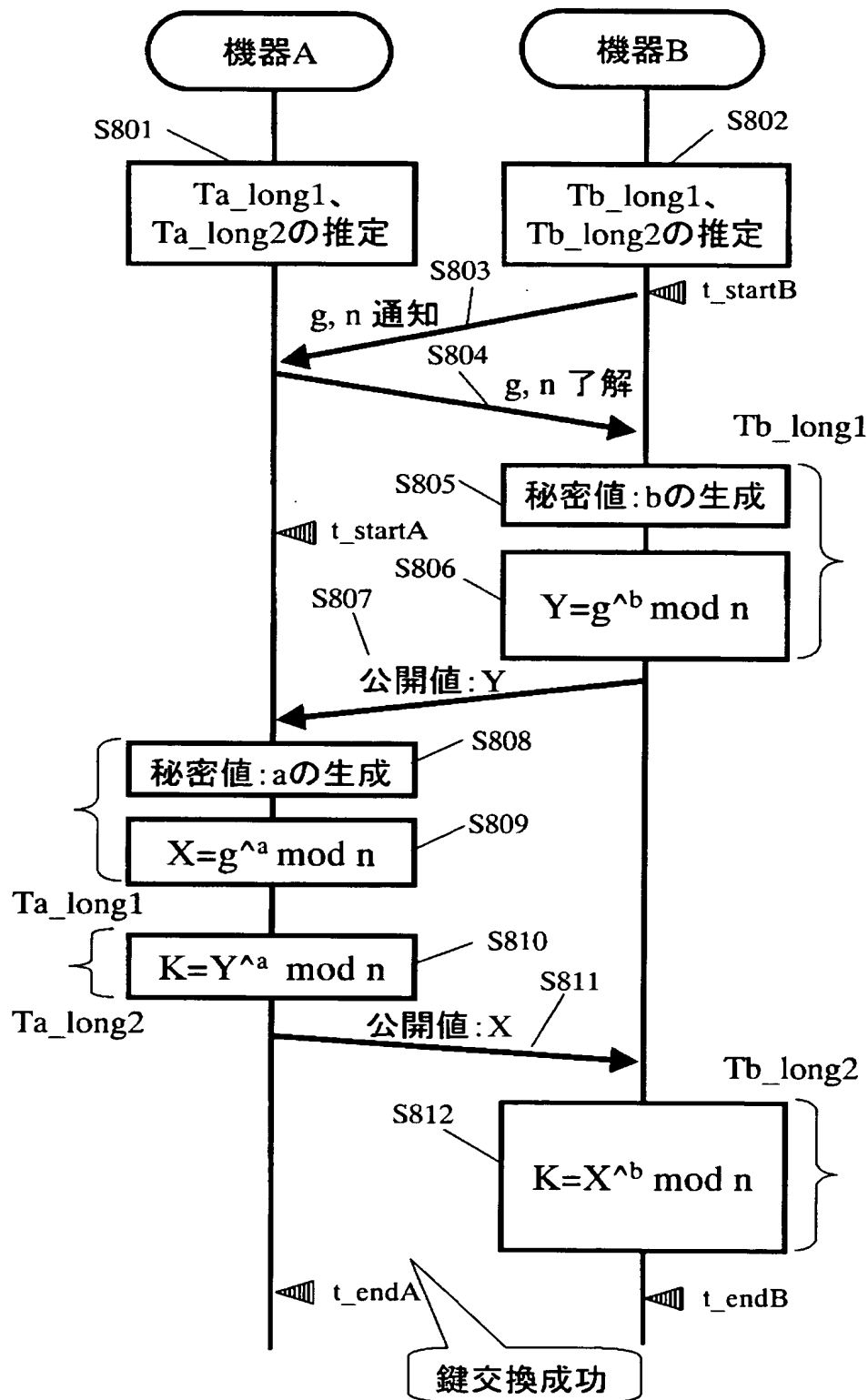
【図6】



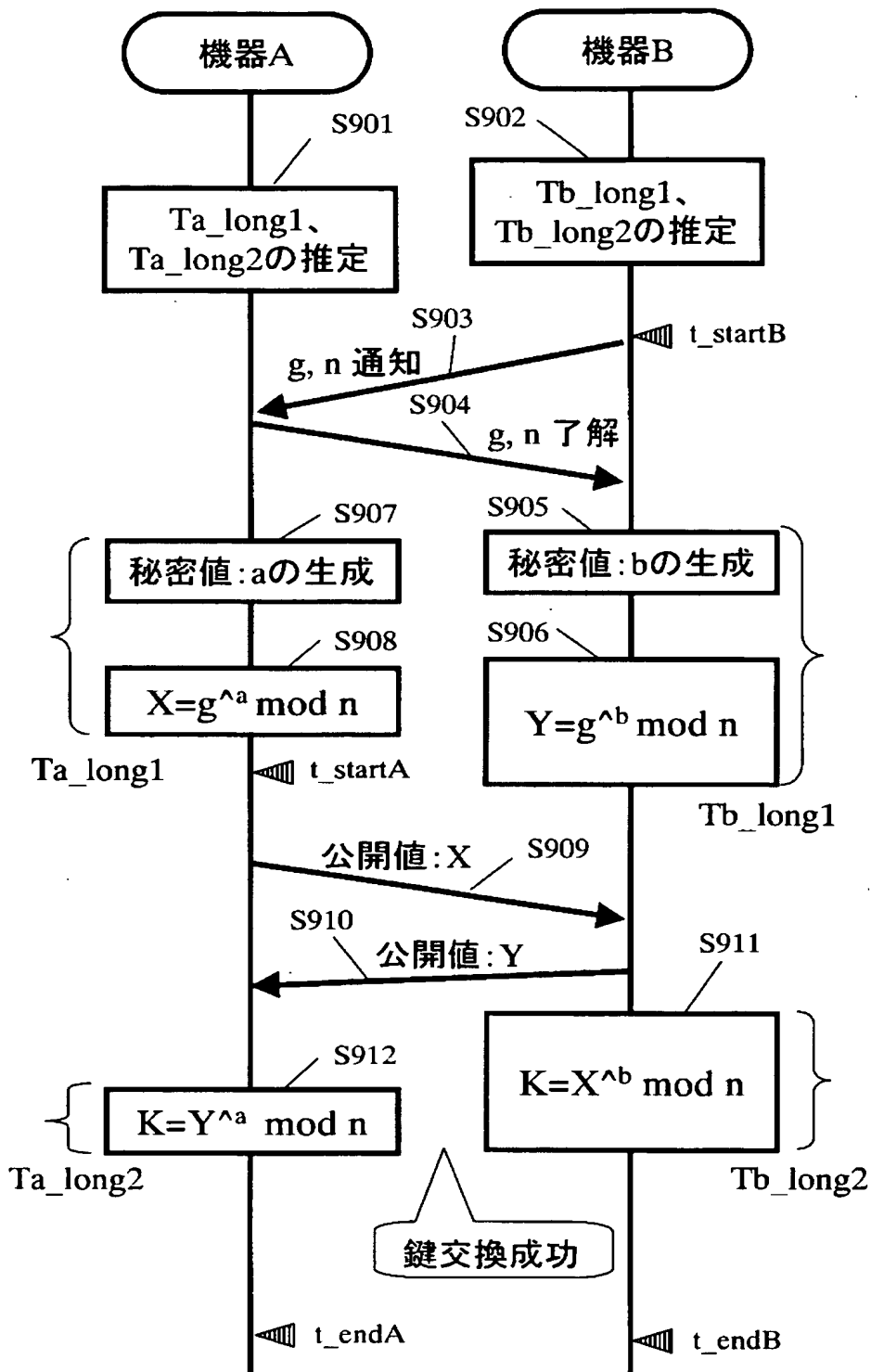
【図 7】



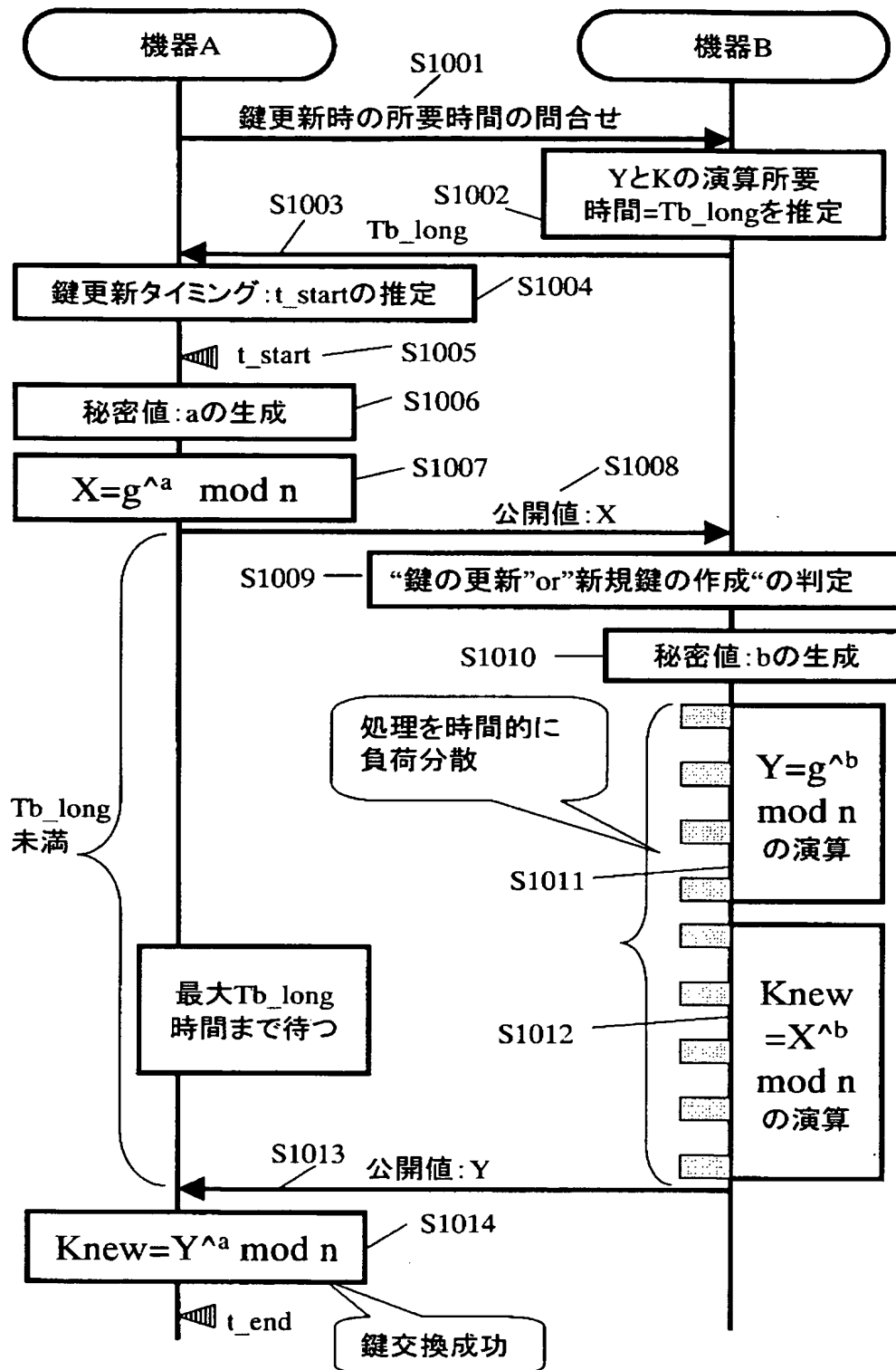
【図 8】



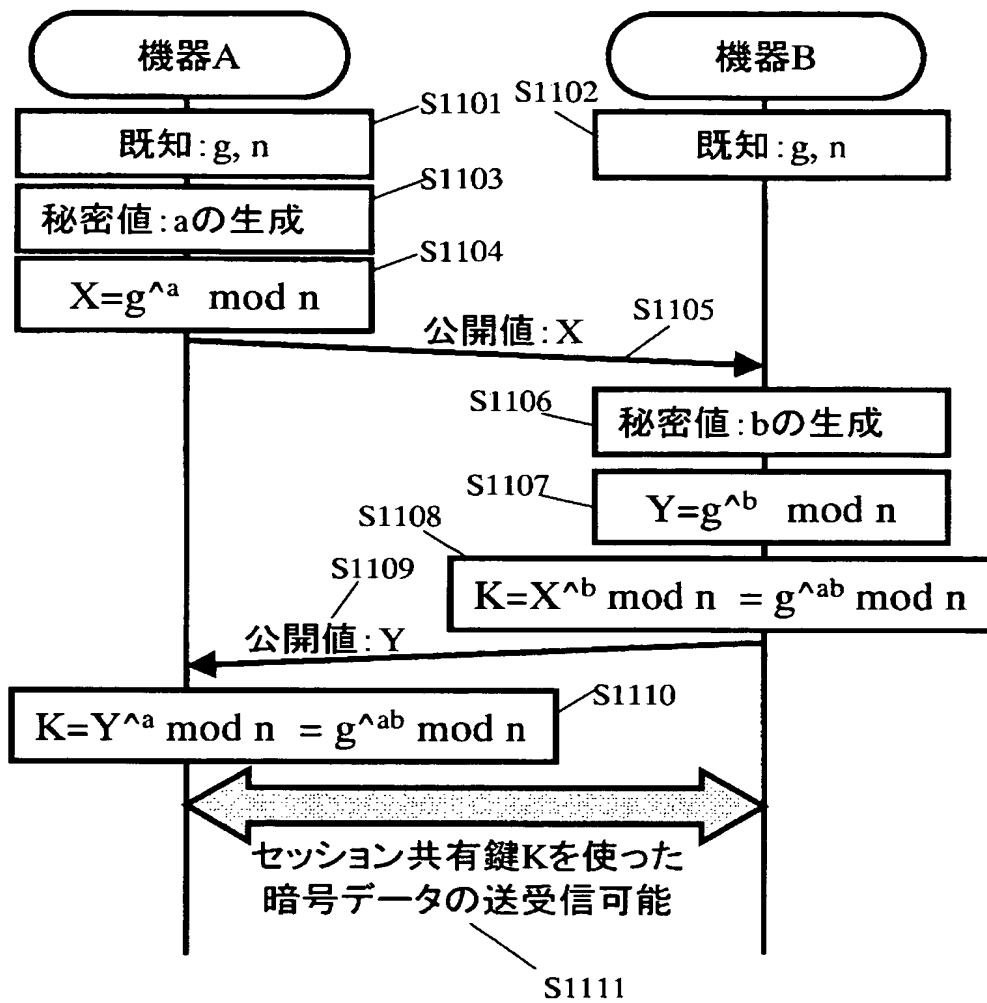
【図 9】



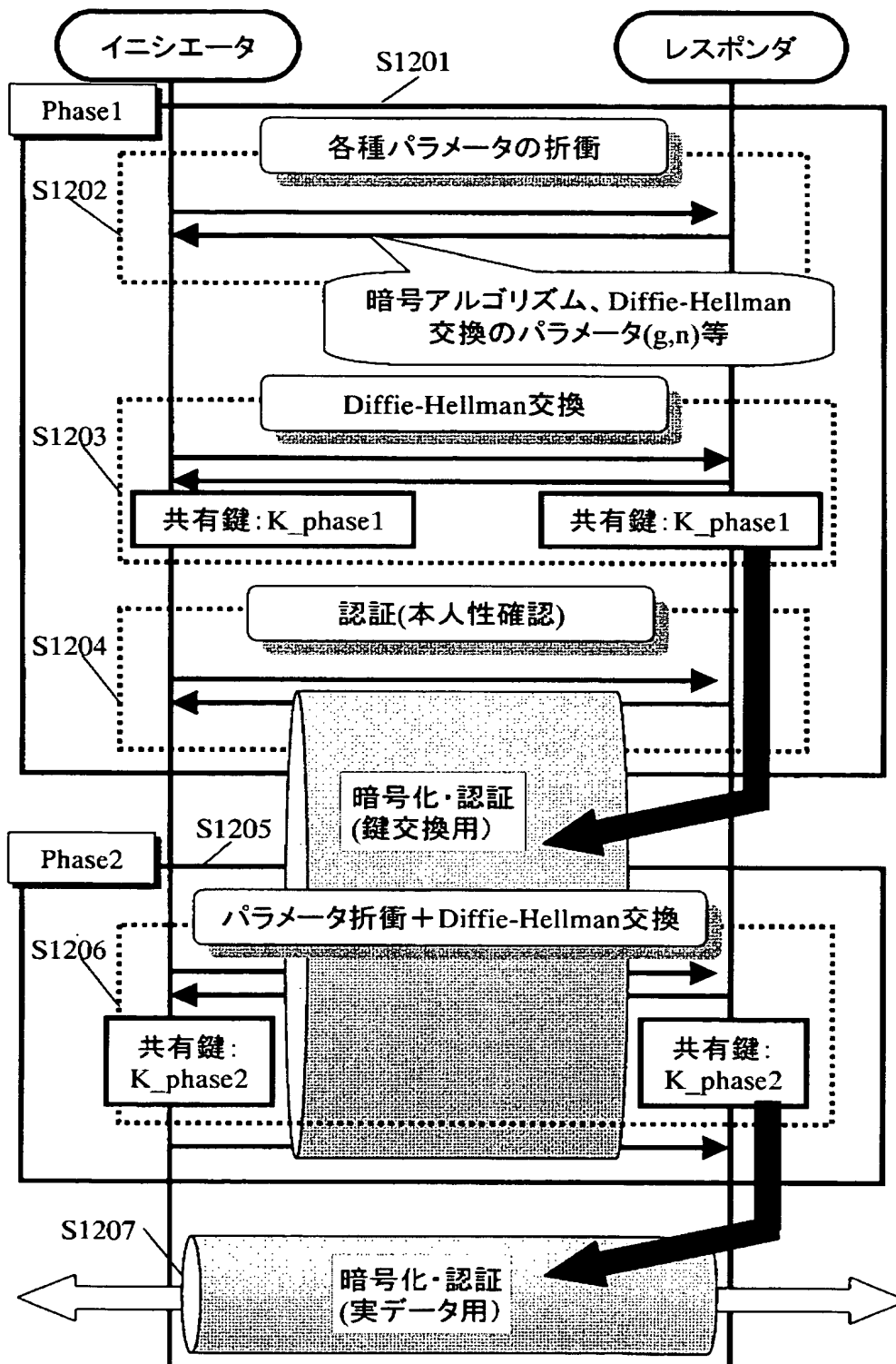
【図10】



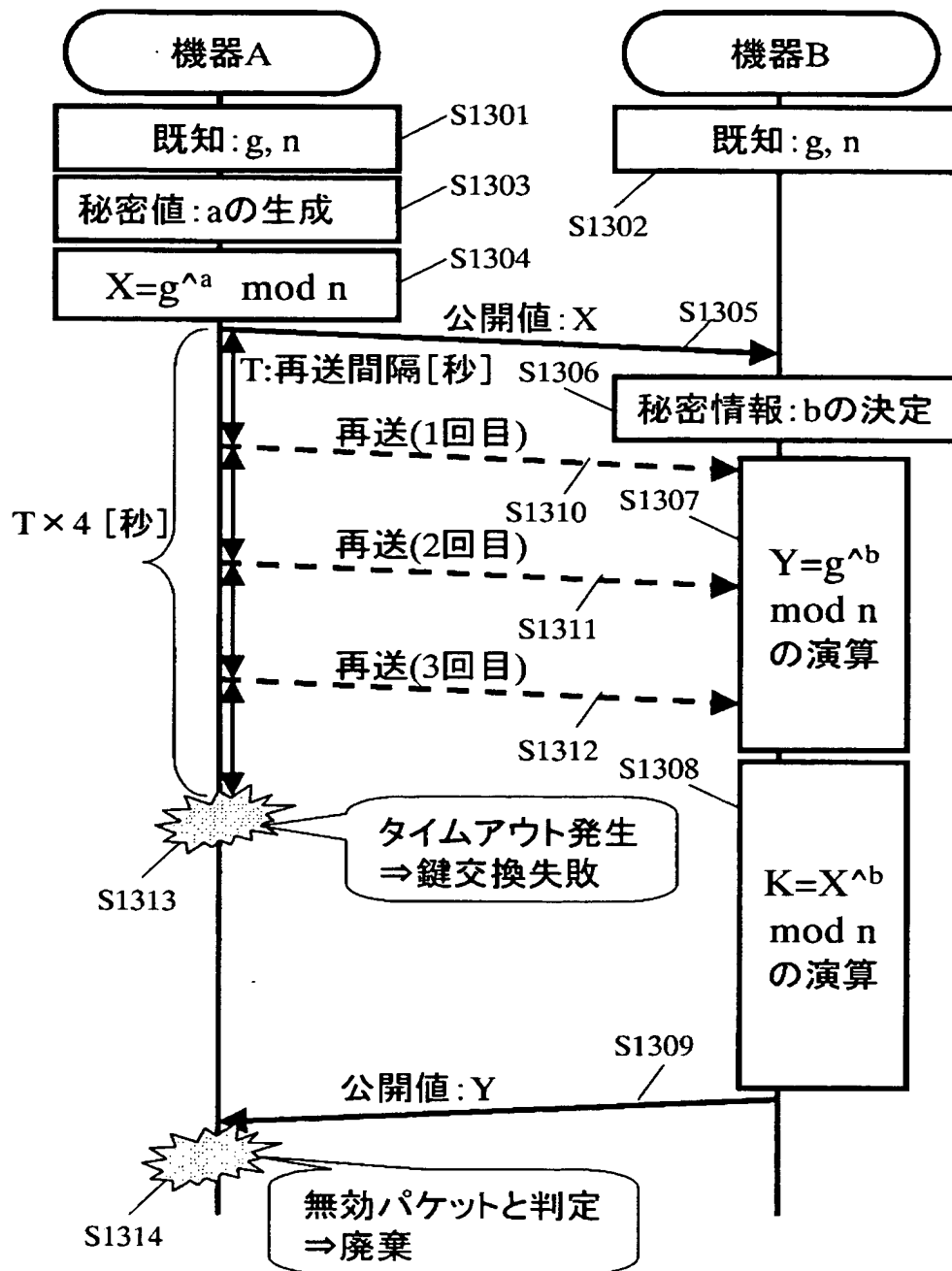
【図 11】



【図 12】



【図 13】



【書類名】 要約書

【要約】

【課題】 2つの通信機器の間で暗号化・認証処理されたデータを送受信するシステムにおいて、低性能のCPUを搭載する機器で、鍵交換に伴う高負荷の演算を行う場合にも、対向の機器側でタイムアウトを検出させることなく鍵交換を成功させるような共有鍵交換方法を提供する。

【解決手段】 低性能のCPUを搭載した機器Bは、機器Aから公開値Xを受信する（S105）と、自分が相手に応答を返すことのできる応答遅延時間：T_bを推定し、機器Aに応答遅延時間を予告する（S107）。その後、機器Bは鍵交換のための演算を行い、公開値Yを算出後機器Aに送信する（S112）。

【選択図】 図1

特願 2 0 0 3 - 0 1 5 8 6 6

出 願 人 履 歷 情 報

識別番号

[0 0 0 0 0 5 8 2 1]

1 . 変 更 年 月 日

1 9 9 0 年 8 月 2 8 日

[変 更 理 由]

新 規 登 録

住 所

大 阪 府 門 真 市 大 字 門 真 1 0 0 6 番 地

氏 名

松 下 電 器 産 業 株 式 会 社